IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	SHEPHERD, James et al.	Examiner:	ROBERTS, Jessica M.
Serial No.:	10/817,436	Group Art Unit:	2621
Filed:	April 5, 2004	Confirmation No.	1291
Title:	APPARATUS AND PROCESS FOR RE-TIMING VIDEO CUTS		

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

I. REAL PARTY IN INTEREST

The real party in interest in this Appeal is Snell & Wilcox Limited, a UK company that is the assignee of the above-identified application.

II. RELATED LITIGATIONS, APPEALS AND INTERFERENCES

There are no prior or pending litigations, appeals or interferences known to Applicants that may be related to, may directly affect, or may be directly affected by or have a bearing on the Board's decision in this Appeal.

III. STATUS OF CLAIMS

Claims 1-12 are pending in this application and are rejected. Applicants have herein canceled claims 5-8 and 12 and are appealing the rejections of claims 1-4 and 9-11.

IV. STATUS OF AMENDMENTS

Applicants have not filed any amendment filed subsequent to the November 24, 2008 final Office Action from the U.S. Patent and Trademark Office. However, in order to

narrow the issues for appeal, Applicants have herein canceled claims 5-8 and 12, leaving only claims 1-3 and 9-11 pending for appeal.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 relates to a process resulting in a sequence of output fields that are interlaced, in which a cut is positioned at a frame boundary, and having the same number of fields as the sequence of input fields. The video process comprises the steps of (1) receiving an interlaced sequence of input fields organized in a plurality of frames; (2) identifying a cut between first and second input fields; (3) identifying whether the cut occurs at a frame boundary; and (4) where a cut occurs otherwise than at a frame boundary, generating from said second field a synthetic field and replacing said first field by said synthetic field. (See page 2, line 30 - page 3, line 5; page 3, lines 15-23; page 6, line 27 - page 7, line 25; Figures 2b, 2c, 2d and 2e.)

Independent claim 9 also relates to a process resulting in a sequence of output fields that are interlaced, in which a cut is positioned at a frame boundary, and having the same number of fields as the sequence of input fields. The video process comprises the same steps (1) - (3) as in claim 1, expect step 4 is (4) where a cut occurs otherwise than at a frame boundary, retiming the cut. (See page 2, lines 26-29; page 4, lines 3-10; Figures 2b, 2c, 2d and 2e.)

VI. GROUNDS OF REJECTIONTO BE REVIEWED ON APPEAL

A. Whether claims 1-4 and 9 are unpatentable under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 6,731,684 (Wu et al.).

B. Whether claims 10 and 11 are unpatentable under 35 U.S.C. § 103 as being unpatentable over U.S. Patent No. 6,731,684 (Wu et al.) in view of U.S. Patent Application Publication No. 2003/0193614 (Holland et al.).

VII. ARGUMENT

A. Background Relevant to the Claims

Television represents moving images by reproducing a sequence of "still" images, each of which represents a temporal sample of an original moving image. The widely-used 525-line and 625-line television systems use "interlaced" scanning formats, in which bandwidth is saved by dividing the scanning lines that make up a complete image into two cyclically repeating groups and by sending only one of these groups for each temporal sample. Specifically, two fields make up a frame (or a picture), with one field having the even numbered lines, and the other field having the odd numbered lines.

A number of recently-designed television formats (referred to as "progressive") do not use interlaced format but rather provide all image lines at each temporal sample. However, some transmission systems for these progressive formats divide the image into two fields to improve compatibility with current equipment designed for interlaced formats and define the field dominance, i.e., the order in which the two fields are transmitted. There are some situations in which it is critical that field dominance be correctly observed and where errors can lead to unacceptable video processing artifacts.

The simplest source of field dominance errors is where a cut between two image sequences incorrectly occurs between the two fields of the same frame. In this case, only half the lines will be available for the last image of the first sequence, and only half the lines will be available for the first image of the second sequence.

The way in which a field dominance error can arise from incorrect editing can be illustrated graphically by the diagrams presented in Appendix II, Tab C. As shown in Diagram 1A, when a cut from Scene A to Scene B is correctly positioned between two frames, field dominance is preserved in the output. However, as shown in Diagram 1B, when a cut from Scene A to Scene B is incorrectly positioned within a frame, a field dominance error is introduced into the output. This ERROR FRAME output contains one field ($A2_{odd}$) for the end of Scene A and one frame ($B2_{even}$) from the start of Scene B. While this sequence will not appear objectionable when viewed field by field, any frame

based processing (including processing to convert to a progressive format) will treat these two likely very different fields as if they were the odd and even lines of a single picture, and the consequences can be highly objectionable to the viewer.

Known methods of correcting field dominance errors in a television signal involved manually editing a recorded version of the program to locate cuts and then either removing incorrectly interlaced frames or replacing a defective frame by a repeat of a complete existing frame. Obviously deletion of frames changes the length of the program and may necessitate changes to the accompanying soundtrack. In any case, both of these manual repair techniques are very time consuming.

B. Claims 1-4 and 9 are not Anticipated by Wu et al

1. The Steps of Independent Claims 1 and 9

The video processes as recited in independent method claims 1 and 9 both require the steps of (1) receiving an interlaced sequence of input fields organized in a plurality of frames; (2) identifying a cut between first and second input fields; and (3) identifying whether the cut occurs at a frame boundary. As can be seen from Figure 2a of the application, a cut that occurs within a frame (see Frame 2) is not at a frame boundary, whereas a cut that occurs between frames is at a frame boundary.

These steps can also be understood graphically by Diagrams 2A and 2B as presented in Appendix II, Tab C. In both Diagrams 2A and 2B, an interlaced sequence of input fields ($A1_{odd}$, $A1_{even}$, $B2_{odd}$, $B2_{even}$, etc.) organized in a plurality of frames (FRAME 1, FRAME 2, etc.) is received. In Diagram 2A, a cut (from Scene A to Scene B) between a first input field $A1_{even}$ and a second input field $B2_{odd}$ can be identified as occurring at a frame boundary (the boundary between FRAME 1 and FRAME 2). In Diagram 2B, the cut between a first input field $A2_{odd}$ and a second input field $B2_{even}$ can be identified as <u>not</u> occurring at a frame boundary. Claims 1 and 9 handle the cut that does not occur at a frame boundary in different ways.

Independent claim 1 requires, where a cut occurs otherwise than at a frame boundary, the step of (4) generating from said second field a synthetic field and replacing said first field by said synthetic field, the process thereby outputting a interlaced sequence of output fields in which the out is positioned at a frame boundary, the sequence of output fields containing the same number of fields as the sequence of input fields. Figures 2b, 2c, 2d and 2e of the application illustrate generation of a synthetic field from one of the fields at either side of the cut and replacing that field with the synthetic field, resulting in the same number of fields.

This step can also be understood by Diagrams 3A and 3B of Appendix II, Tab C. As shown in Diagram 3A, the interlaced sequence of Diagram 2B where the cut is "otherwise than at a frame boundary" is modified. A synthetic field $B2_{synthetic}$ is generated from the "second field" $B2_{even}$, and the "first field" $A2_{odd}$ is then replaced by synthetic field $B2_{synthetic}$ to produce (as shown in Diagram 3B) an interlaced sequence of output fields in which the cut is positioned at a frame boundary and the sequence of output fields contains the same number of fields as the sequence of input fields. It should be noted that $B2_{even}$ could be the "first field", and $A2_{odd}$ could be the "second field".

Independent claim 9 requires, where a cut occurs otherwise than at a frame boundary, the step of (4) retiming the cut, the process thereby outputting a interlaced sequence of output fields in which each cut is positioned at a frame boundary, the sequence of output fields containing the same number of fields as the sequence of input fields. Figures 2b, 2c, 2d and 2e of the application illustrate that the new cut is on either side of Frame 2, not within it.

This step can also be understood by Diagrams 3A, 3B and 4 of Appendix II, Tab C. As shown in Diagram 4, once the synthetic field $B2_{synthetic}$ is generated from the "second field" $B2_{even}$ and has replaced the "first field" $A2_{odd}$, the timing of the cut is changed so to position the cut at a frame boundary, i.e., from between $A2_{odd}$ and $B2_{even}$ to before $B2_{synthetic}$ (which replaced $A2_{odd}$). As a result, the interlaced sequence of output fields and the sequence of output fields contains the same number of fields as the sequence of input fields.

2. U.S. Patent No. 6,731,684 (Wu et al.)

Wu et al. discloses method and apparatus for detecting scene changes and adjusting picture coding type in a HDTV encoder. The invention of Wu provides an encoding architecture that enables I-frames, which start a new Group of Pictures (GOP), to be aligned with scene changes between successive fields.

a. Background Relevant to Wu

Wu is concerned with the MPEG2 compression scheme for video, which brings together a range of compression techniques. One of these techniques is temporal compression, which is effective with video material because there is frequently a close similarity between one image and the next succeeding image in the video sequence. In temporal compression, the current picture is not sent in its entirety; instead, since the decoder already has the previous picture, only the difference between the current picture and the previous picture is sent, so that the decoder can add the difference and make the current picture.

Where a video sequence portrays motion, there may still be a large difference between one image and the next, and, thus, little compression. MPEG2 addresses this issue by taking a motion compensated difference between adjacent images. Simply put, motion compensation tracks the movement of objects portrayed in the video sequence so that – effectively – objects can be put back to where they would have been in the absence of motion, before the difference is taken. Since it is necessary to give the decoder only a description of the tracked motion of the objects, which is done with motion vectors, this results in a much smaller difference signal and, thus, greater compression.

In the nomenclature adopted in MPEG2, an entire picture is sent first in the form of an I-frame, wherein the "I" represents "intra" coding of the frame, i.e., no temporal compression is utilized in that frame. As described above, only the differences between one picture and the next picture are then sent, and these differences are sent as P-frames, wherein the "P" stands for "predictive" coding (since the decoder has available to it the last

picture, which can be regarded as a "prediction" of what the next picture will be). The difference information received in the P-frame enables the decoder to generate the next picture by applying the differences to its prediction. Where motion compensated difference information is provided, the decoder forms its prediction by taking the last picture and moving the objects depicted therein in accordance with the motion vectors.

However, it is not practical to continue sending only difference information. One reason is that small and unavoidable errors in the coding process will grow with time, with a steady decline in quality of the decoded picture. Another reason is that most video sequences are made up of a series of shots with cuts being made between the output of different cameras, wherein the last picture in one shot may bear no relationship to the first image of the next shot. Thus, predicting the first shot of a new sequence using the last shot of the old sequence makes no technical sense. For these reasons it is usual, where a predictive coding is employed, to send a complete picture (I-frame) at regular intervals.

Typically, an encoder will have a coding strategy that specifies how often an I-frame is to be sent. For example, an MPEG2 compressed signal might consist of a sequence: I P P P I P P P I P P P, i.e., an I frame is sent for every three P frames that are sent. Each group, starting with an I-frame and including all the succeeding P-frames, is referred to as a Group of Pictures (GOP). It is important to understand that, unlike an I-frame, a P-frame is not a picture at all but rather is only a set of instructions to convert the previous picture, as defined by the most recent I frame and its succeeding P frames, into the current picture.

Where a decoder has a fixed and unchanging strategy, it may be thought of as having a fixed GOP structure. Other decoders may choose to adapt their GOP structure depending upon the content of the video signal being encoded, such as by seeking to identify cuts in the video sequence and to ensure that the first picture in a new shot is always encoded as an I-frame. By way of illustration, see the following example:

Frame:	1	2	3	4	5	6	7	8	9	10
Video sequence with cut between shot A and shot B:	A	A	A	B	B	B	B	B	B	B
Output of coder with fixed GOP structure:	I	P	P	P	P	I	P	P	P	P
Output of coder with cut detection and variable GOP structure:	I	P	P	I	P	P	P	P	I	P

In the case of a coder with a fixed GOP structure, video frame 4 is encoded as a P frame, even though it is in fact the start of the new shot B. By contrast, a coder with a cut detector will recognize that video frame 4 is the start of the new shot B and will vary its GOP structure to ensure that frame 4 is encoded as an I-frame.

Clearly, frame 4 in the output of the coder with a cut detector and variable GOP structure, which is an I-frame, i.e., a full picture corresponding to picture B, is very different from frame 4 of the output of the coder with the fixed GOP structure, which is a P-frame, i.e., not a picture, just a set of differences that, essentially, represent the difference between the fourth video frame B and the previous P-frame. That previous P-frame is in turn the difference between the third video frame A and its preceding P-frame, that is again a set of differences, and so on. Thus, the I-frame that is a representation of the fourth video frame B cannot be generated using only the information contained in the P-frame that would have been generated by a coder using a fixed GOP structure.

b. The Disclosure of Wu

Wu discloses a method and apparatus for detecting scene changes between successive fields in a digital television signal. Wu describes an example of a coder with a cut detector and variable GOP structure as described above and discloses building a coder

having a fixed GOP structure, adding a cut detector and adding additional processing to ensure that, taking the simple example set forth above, frame 4 is encoded as an I-frame.

Wu recognized that the coding efficiency of an MPEG encoder can be improved if the leading I-frame in a GOP is aligned with the start of a new scene. The intent of the Wu disclosure is that this final decision to encode a frame as an I-frame or P-frame is made with appropriate information regarding scene changes so that, wherever appropriate, an I-frame is aligned with the start of a new scene. As stated by Wu at column 2, lines 9-14:

> "In accordance with the present invention, scene change detection is performed at a pre-processing stage of a video encoder. The final decision to encode a frame an I- or P-frame is not made until the final encoding stage. That is, the encoders processing pipeline that is used as a lookahead buffer to minimize the amount of required frame buffer memory."

At the top of Figure 2, Wu shows an encoder processing chain comprising pre-processing stage 205, PB frame re-ordering delay 215, motion estimation stage 220 and encoder stage 225. Input video entering the pre-processing stage 205 is compressed in this processing chain and output as an MPEG bitstream. The lower half of Figure 2 contains the functionality, which is Wu's contribution. It is important to note that the circuit elements shown in the lower half of Figure 2 (scene change detection 210, delay 230, scene change counter 240 and picture coding type decision 235) serve to provide only one input to the video compression processing chain. This input is the "picture-type" parameter output by the picture encoding type decision 235 to the encoding stage 225.

The compression processing chain in the top half of Figure 2 is conventional and, in the absence of the "picture type" parameter provided by the additional circuitry of Wu, would take decisions regarding the location of I-frames. Thus, in a long unbroken scene, the location of I-frames would be decided in the compression processing chain. If a scene change is detected, the function of the additional circuitry of Figure 2 may result in a change of decision, such that a frame that would otherwise have been encoded as a P-frame may instead be encoded as an I-frame.

The scene change detection function 210 detects scene changes between consecutive frames at the pre-processing stage of the pipeline so as to change the location of the scheduled start of a new GOP in order to align with the start of a new scene, if a scene change is detected at the proximity of the originally scheduled I-frame. Once a scene change is detected by scene change detection function 210, a control signal ScDet flag is sent to the picture coding type decision function 235. Based on the results of the scene change detection, the encoder's buffer level and the number of frames coded so far in the GOP, the picture coding type decision function 235 makes a final decision for the picture coding type of the frame at the encoding stage 225 of the pipeline, as discussed in Wu at column 5, lines 32-38. By use of the delay 230, once the scene change frame arrives at encoding stage 225, that frame is encoded as an I-frame.

Somewhat confusingly, however, when describing the use of an I-frame (rather than a P-frame) at the first picture in a new shot (see the passage of Wu referred to by the Examiner at columns 9-10), Wu states that "the P-frame is changed to an I-frame". However, Applicants state that Wu in fact intended to state that the decision that would have been taken by a coder having a fixed GOP structure, i.e., a decision to use a P-frame, is changed to a decision to use an I-frame. Thus, the "change" referred to in Wu is a only change in coding decision that takes place before the frame has been compressed as either an I-frame or a P-frame, i.e., a change from what would have occurred if the scene change technology of Wu were not present. In fact, there is no actual change of a P-frame to an I-frame, and the I-frame is no sense generated from the P-frame.

3. Wu Does Not Anticipate Claims 1-4

In rejecting independent claim 1 as being anticipated by Wu, the Examiner stated that Wu teaches the step of "where a cut occurs otherwise than at a frame boundary, generating from said second field a synthetic field and replacing said first field with said synthetic field". According to the Examiner, this disclosure is at column 9, lines 21-25, where Wu states that, when a scene change is detected, "a P frame is converted into an I

frame". The Examiner concluded that "the process thereby outputting a interlaced sequence of output fields in which the cut is positioned at a frame boundary." (See Final Office Action, at page 7.)

Applicants respectfully request that the rejection be reversed, as this rejection is founded upon the Examiner's misunderstanding of the disclosure of Wu. Contrary to the Examiner's statements, as will be discussed below, there is no disclosure in Wu of the generation of a synthetic field of picture information from a second field, and there is no disclosure of the replacement of a first field by the synthetic field, as required by claim 1.

Clearly, P-frames and I-frames exist in the output bitstream. However, Wu does not disclose to take from the output bitstream a frame that has been encoded as a P-frame and to convert it into a frame compressed as I-frame. To the extent that Wu refers to frames of uncompressed video as "I, P or B", this, as would be understood by one of skill in the art, merely denotes the preliminary choice of encoding mode which will be used in the final coding stage on the video frame in question. However, there is no actual transformation in Wu of a P-frame into an I-frame.

Instead, Wu teaches that a preliminary decision as to whether a particular frame is to be encoded as an I-frame or a P-frame or a B-frame made by the pre-processing stage 205 is either confirmed or modified by the picture coding type decision function 235, and a single encoding operation is carried out by the encoding stage 225. Applicants contend that it is not possible as a matter of technology to view the change of coding decision in Wu as the generation of synthetic field (I-frame) from that P-frame. Wu teaches that a preliminary decision in Wu to generate a P-frame is changed into a final decision to generate an I-frame, such that the I-frame in question is generated in place of the P-frame. However, because that P-frame is not generated, the I-frame is not, and therefore could never be, generated from the P-frame, since the P-frame is never actually formed.

Moreover, even if the P-frame had been formed (which it is not), the I-frame could still not have been generated from the P-frame. Since an I-frame is a full picture while a P-frame is only a set of differences, the I-frame requires information that would simply not

be present in the P-frame. As such, even if Wu did teach that the P-frame is changed into an I-frame (which it does not), there simply is insufficient information in a P-frame to allow it to be changed into an I-frame.

Furthermore, even if Wu did teach that the P-frame is changed into an I-frame (which it does not), there is no indication that an I-frame is in any way considered a "synthetic field", as required by independent claim 1. Thus, there is no disclosure or suggestion of the generation of a synthetic field from a second field or of the substitution of a first field by said synthetic field.

In response to Applicants' arguments, the Examiner referred specifically to Wu at column 10, lines 19-22, which stated:

> Generally, when a scene change frame is detected, and the scene change frame is a P-frame, it is changed to an I-frame. If the scene change frame is not a P-frame, the first P-frame following the scene change frame is changed to an I-frame. (Wu, at column 10, lines 19-22).

The Examine then interpreted this passage from Wu by stating as follows:

> Since converting is analogous to change, and Wu discloses that when a scene change frame is detected, and the scene change frame is a P-frame, it is changed to an I-frame, therefore, it is clear to the examiner that changing the P-frame to an I-frame when a scene change is detected, is equivalent to converting the P-frame to an I-frame, which reads upon the claimed limitation. (Final Office Action, at page 7.)

Applicants contend that the Examiner misunderstands what Wu means by the terminology "the P-frame is changed to an I-frame", and the Examiner's rejection of independent claim 1 is founded on this misinterpretation. As discussed above, there is no physical or even electronic conversion of a P-frame into an I-frame in Wu. Rather, Wu clearly means that a frame which in the absence of the scene change technology of Wu would have been encoded as a P-frame is instead fact encoded as an I-frame. The "change" has in fact nothing whatsoever to do with the creation of a synthetic field as proposed in the present invention.

As stated above, the "change" in Wu is actually a change in coding decision, which takes place before the frame has been compressed as either an I-frame or a P-frame. This

"change" is in fact not an actual change but rather a change in intent, i.e., a change from what would have occurred if the scene change technology of Wu were not present. In fact, the P-frame (from which the Examiner contends a synthetic frame is generated) never actually exists, and, even if it did, it is technically not possible to change that P-frame into an I-frame because the P-frame does not contain sufficient information for the generation of the I-frame.

Accordingly, Applicants respectfully assert that the rejection of claim 1 should be reversed, as Wu cannot alone anticipate the invention of independent claim 1. Claims 2-4, which depend from independent claim 1 and include all the limitations thereof, are likewise not anticipated by Wu for the same reasons, Applicants respectfully request that the rejection of claims 2-4 be reversed as well.

4. Wu Does Not Anticipate Claim 9

In rejecting independent claim 9 as being anticipated by Wu, the Examiner states that Wu teaches the step of "where a cut occurs otherwise than at a frame boundary, retiming the cut". According to the Examiner, this disclosure is at column 4, lines 43-49, where Wu allegedly discloses a delay function that accounts for delays in processing the corresponding frame in the delay function and motion estimation. The Examiner concludes that, "since the delay accounts for delays in processing, it would be capable of re-adjusting or retiming a scene change, the process thereby outputting a interlaced sequence of output fields in which the cut is positioned at a frame boundary." (See Final Office Action, at page 9.)

At the referenced passage, Wu states as follows:

> In particular, the scene change flag is provided to a delay 230 to account for the delays in processing the corresponding frame in the reordering delay function 215 and motion estimation stage 220, and to a picture coding type decision function 235. The function 235 sends a corresponding picture_type control signal to the encoding stage 225 to set the picture type of the current frame to be encoded. (Wu, column 4, lines 43-49.)

Thus, in alleging that Wu discloses "where a cut occurs otherwise than at a frame boundary, retiming the cut", the Examiner refers to delay 230 in Figure 2 of Wu. The Examiner states that the purpose of delay 230 is to compensate for the delay, particularly, blocks 215 and 220 of the compression processing chamber, and states that the delay 230 would be capable of re-adjusting or re-timing a scene change.

Applicants respectfully request that the rejection be reversed, as this rejection is founded upon the Examiner's misunderstanding of the disclosure of Wu. Applicants contend that this statement by the Examiner lacks any foundation in Wu and is directly contrary to the expressed teaching of Wu. The purpose of delay 230 in Wu is to ensure that the picture coding decision is in precise register with the scene change.

By contrast, the re-timing of a cut as recited in claim 9 operates so that, where the input fields exhibit a cut in the middle of a frame, the cut is re-timed such that the cut is instead positioned at a frame boundary in the output fields. Wu does not disclose this processing and indeed the structure described in Wu is incapable of such processing. As further confirmation of the fact that Wu does not operate to re-time cut that occurs in the middle of a frame, Applicants note that Wu proposes an alternative approach to addressing the issue of cuts within a frame -- at column 2, lines 43-46, Wu proposed that, in the case of a "bad edit" where a scene change occurs at the odd/even field boundary of a frame, the step is taken of switching from frame prediction to field prediction.

In addition, Applicants point out that, in the Examiner's rejection of claim 5 at page 11, lines 19-21 of the final Office Action, the Examiner stated that "Wu fails to teach the input sequence being automatically retimed to occur at a frame boundary in the output sequence." Applicants point out that the Examiner has understated the fact that, not only does Wu not teach that the input sequence is automatically retimed to occur at a frame boundary in the output sequence, but that Wu does not teach any retiming of the cut, where the cut occurs otherwise than at a frame boundary.

Accordingly, Applicants respectfully assert that the rejection of claim 9 should be reversed, as Wu cannot alone anticipate the invention of claim 9.

C. Claims 10 and 11 are Not Obvious over Wu et al. in view of Holland et al.

1. The Steps/ Elements of Claims 10 and 11

Both of claims 10 and 11 are dependent upon independent claim 9, as discussed above. Claim 10 additionally recites that the step of retiming comprises generating a synthetic field through motion compensation. Claim 11 additionally recites that the step of retiming comprises generating a synthetic field through interpolation.

2. U.S. Patent Application Publ. No. 2003/0193614 (Holland et al.)

Holland et al. discusses detecting and correcting motion artifacts in interlaced video signal converted for progressive video display, i.e., detecting scene changes and adjusting picture coding type in a HDTV encoder. A correction is applied where interlaced video material is determined to originate from film source, thereby having been converted to video using a process known as 3-2 pulldown. Where the video material is not a result of the 3-2 pulldown process, a check is made for the presence of "pixel motion" so that corrections may be applied to smooth out the pixel motion. To determine 3-2 pulldown or field motion, a video field is compared to the field prior to the previous field to generate field error. Field errors are generated for five consecutive fields and a local minimum error repeated every five fields indicate the origination of the video material from film source using the 3-2 pulldown process. Upon confirmation of 3-2 pulldown, the video material is modified to correct for the mixing of two film frames into one interlaced video frame by assuring that the two fields of the de-interlaced video frame contain data from the same film frame. Where the video material did not originate from a film source, but pixel motion is detected, the pixel motion is smoothed out by an averaging method. The odd and even fields of the resulting video data are subsequently combined to form a progressive video material. (Holland et al., Abstract)

3. Wu and Holland Do Not Render Claims 10 and 11 Obvious

Applicants submit that one of ordinary skill in the art would not combine Wu et al. with Holland et al. Wu et al. relates to the selection of an I-frame position dependent on a scene-change position in a video compression coder, and provides an encoding architecture that enables I-frames, which start a new Group of Pictures (GOP), to be aligned with scene changes between successive fields. By contrast, Holland et al. relates to methods and apparatus for correction of 2-3 field patterns and refers to the position of fields within a sequence of fields. There is no motivation for one of ordinary skill in the art to combine the teachings of these two documents.

Even if Wu et al. and Holland et al. were combined, the invention of claims 10-11 would not be rendered obvious. In particular, as discussed above, Wu et al. does not show the claimed step of "where a cut occurs otherwise than at a frame boundary, retiming the cut", as recited in claim 9 upon which claims 10 and 11 are dependent. In the apparatus of Wu et al., no consideration is given to the frame structure of the interlaced fields. In fact, the arrangement in Wu et al. merely detects the frame at which scene change occurs in order to alter the position of an I-frame and does not retime the cut.

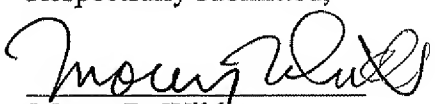
Holland et al. does not solve the deficiencies of Wu et al. with respect to independent claim 9. In Holland et al., a disrupted video signal is analyzed to generate correction information and used to correct the disrupting video signal, resulting in undisrupted video signal with continuous 2-3 field sequence. However, there is no disclosure in Holland et al. of the feature of automatic re-timing of the cut to occur at a frame boundary, as recited in claim 9.

Thus, claims 10-11, which are dependent upon independent claim 9 and require all the steps thereof, are likewise not obvious based upon Wu et al. and Holland et al., and Applicants respectfully request that this rejection be reversed.

VII. Conclusion

The Office is authorized to charge the fee for the Appeal Brief of $500.00 under 37 C.F.R. § 41.20(b)(2) to deposit account No. 50-3355. No other fees are believed to be due in connection with this paper. However, if such additional fees are due, please charge any such fees to deposit account No. 50-3355.

Respectfully submitted,



Morey B. Wildes
Attorney/Agent for Applicant(s)
Registration No. 36,968

Dated: July 23, 2009

Pearl Cohen Zedek Latzer, LLP
1500 Broadway, 12th Floor
New York, New York, 10036
Tel: (646) 878-0800
Fax: (646) 878-0801

APPENDIX I: CLAIMS

1. (Previously Presented) A video process comprising the steps of

receiving an interlaced sequence of input fields organized in a plurality of frames;

identifying a cut between first and second input fields;

identifying whether the cut occurs at a frame boundary; and,

where a cut occurs otherwise than at a frame boundary, generating from said second field a synthetic field and replacing said first field by said synthetic field,

the process thereby outputting a interlaced sequence of output fields in which the cut is positioned at a frame boundary, the sequence of output fields containing the same number of fields as the sequence of input fields.

2. (Original) A process according to claim 1, wherein the second field appears after the first field in the temporal sequence.

3. (Original) A process according to claim 1, wherein the step of generating a synthetic field from said second field, comprises a step of motion compensation such that objects represented in said second field are positioned in said synthetic field at the locations they are estimated to occupy at the time associated with said first field.

4. (Original) A process according to claim 1, wherein the step of generating a synthetic field from said second field, comprises a step of interpolation such that objects represented in said second field are positioned in said synthetic field with the vertical positioning associated with said first field.

5-8. (Canceled)

9. (Previously Presented) A video process comprising the steps of

receiving a sequence of input fields organized in a plurality of frames;

identifying a cut between first and second input fields;

identifying whether the cut occurs at a frame boundary; and,

where a cut occurs otherwise than at a frame boundary, retiming the cut,

the process thereby outputting a interlaced sequence of output fields in which each cut is positioned at a frame boundary, the sequence of output fields containing the same number of fields as the sequence of input fields.

10. (Previously Presented) A process according to claim 9, wherein the step of retiming comprises generating a synthetic field through motion compensation.

11. (Original) A process according to claim 9, wherein the step of retiming comprises generating a synthetic field through interpolation.

12. (Canceled)

APPENDIX II: EVIDENCE

Tab	Evidence
A.	U.S. Patent No. 6,731,684 (Wu et al.), cited by the Examiner.
B.	U.S. Patent Application Publication No. 2003/0193614 (Holland et al.), cited by the Examiner.
C.	Illustrative diagrams regarding background of the invention.

US006731684B1

(12) **United States Patent**
Wu

(10) **Patent No.: US 6,731,684 B1**
(45) **Date of Patent: May 4, 2004**

(54) **METHOD AND APPARATUS FOR DETECTING SCENE CHANGES AND ADJUSTING PICTURE CODING TYPE IN A HIGH DEFINITION TELEVISION ENCODER**

(75) Inventor: **Siu-Wai Wu**, San Diego, CA (US)

(73) Assignee: **General Instrument Corporation**, Horsham, PA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/806,390**

(22) PCT Filed: **Aug. 27, 1999**

(86) PCT No.: **PCT/US99/19804**

§ 371 (c)(1),
(2), (4) Date: **Mar. 29, 2001**

(87) PCT Pub. No.: **WO00/19726**

PCT Pub. Date: **Apr. 6, 2000**

Related U.S. Application Data

(60) Provisional application No. 60/102,234, filed on Sep. 29, 1998.

(51) **Int. Cl.**[7] **H04N 7/12**
(52) **U.S. Cl.** **375/240.12**; 348/699
(58) **Field of Search** 348/699, 700, 348/701, 419.1; 375/240.12, 240.26; 382/232, 233, 242; H04N 7/12

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,343,248	A		8/1994	Fujinami
5,500,689	A		3/1996	Lam
5,508,750	A		4/1996	Hewlett
5,686,963	A	*	11/1997	Uz et al. 375/240.06
5,734,419	A		3/1998	Botsford, III et al.
5,751,378	A		5/1998	Chen et al.
5,771,316	A		6/1998	Uz
5,832,121	A		11/1998	Ando
5,899,575	A		5/1999	Okayama

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

EP	0554086	8/1993
WO	0708564	4/1996
WO	9739577	10/1997

OTHER PUBLICATIONS

"Scene change detection in GA encoder", published in digitalHDTV Grand Alliance System Record of Test Results, Jun. 28, 1995.

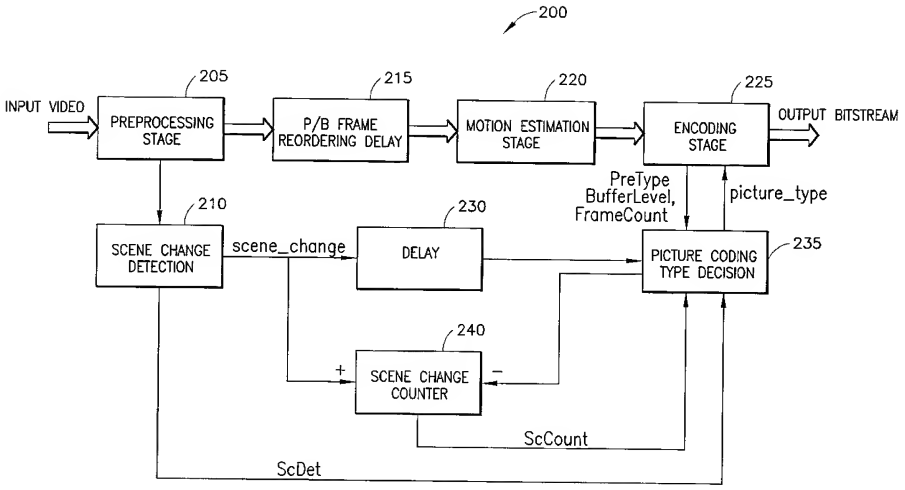
(List continued on next page.)

Primary Examiner—Nhon Diep
(74) *Attorney, Agent, or Firm*—Barry R. Lipsitz; Douglas M. McAllister

(57) **ABSTRACT**

A method and apparatus are provided for detecting scene changes between successive fields in a digital television signal. I-frames, which start new GOPs, are aligned with scene changes. In a preprocessing stage (**205**, **210**), the change in the sum of pixel differences between consecutive odd fields, or consecutive even fields, is calculated for every consecutive input field. A scene change is detected when a large positive value in the change of sum is followed by a large negative value therein. A decision of which picture type to use is not made until a final encoding stage (**225**, **235**). I-frames can be inhibited when an encoder buffer level is too high. A counter resets the scene change indication to avoid a perpetual scene change state for transitions from still to motion. In film mode, the MPEG-recommended frame-based encoding is deactivated when a scene change occurred on a field boundary (e.g., when there is no redundant field in the picture).

21 Claims, 7 Drawing Sheets



U.S. PATENT DOCUMENTS

5,909,252	A		6/1999	Ueda et al.	
5,923,376	A		7/1999	Pullen et al.	
6,044,115	A		3/2000	Horiike et al.	
6,084,641	A		7/2000	Wu	
6,091,776	A	*	7/2000	Linzer	375/240.12
6,408,024	B1	*	6/2002	Nagao et al.	375/240.01
6,434,196	B1	*	8/2002	Sethuraman et al.	375/240.12
6,587,506	B1	*	7/2003	Noridomi et al.	375/240.12

OTHER PUBLICATIONS

Advanced Television Systems Committee Standard A/54, "Guide to the Use of the ATSC Digital Television Standard", Oct. 4, 1995.

ISO/IEC 13818–2, "Generic Coding of Moving Picutures and Associated Audio: Video", (MPEG2 Standard) 1995 E.

* cited by examiner

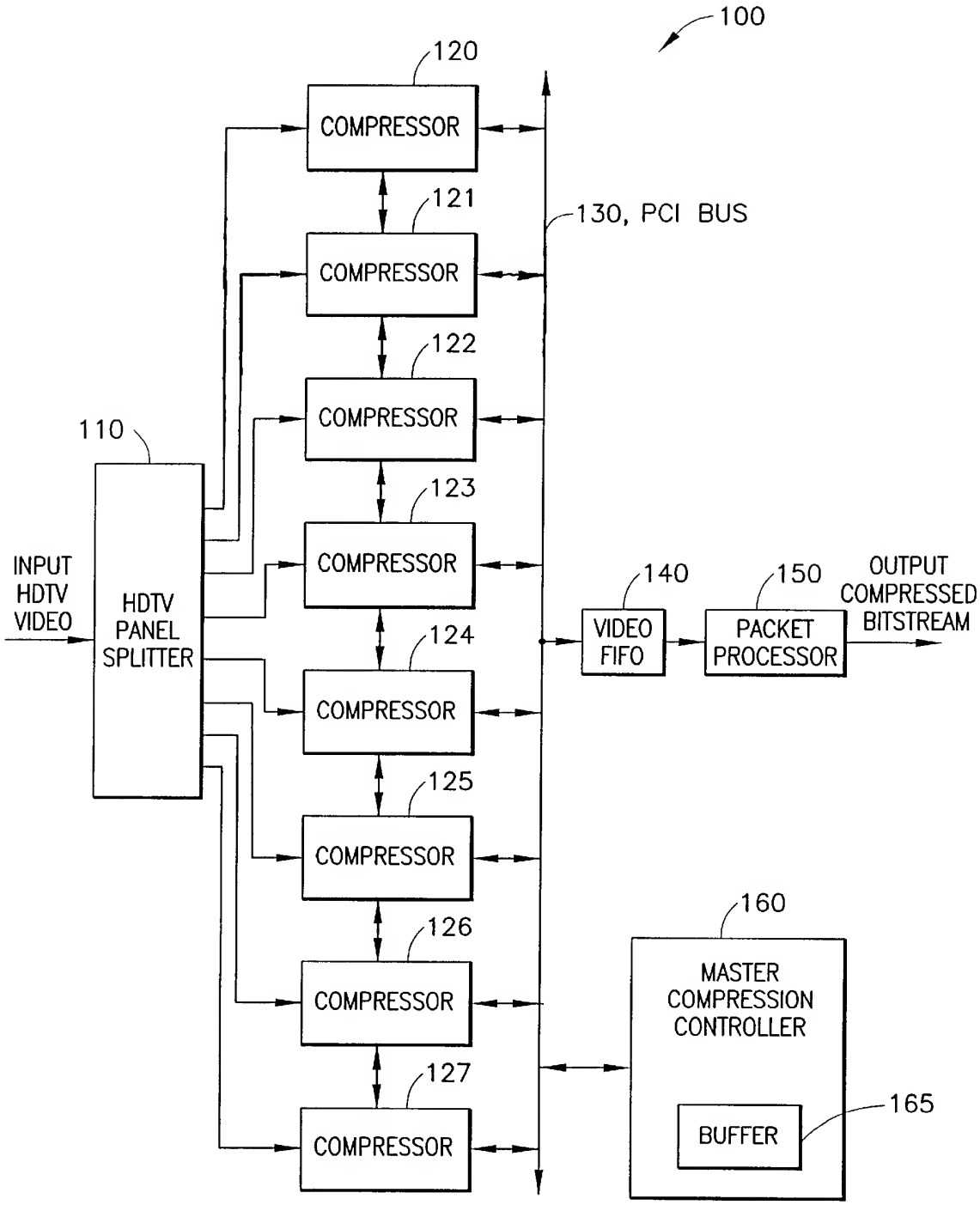

100
120
COMPRESSOR
121
COMPRESSOR
130, PCI BUS
122
COMPRESSOR
110
123
COMPRESSOR
INPUT HDTV VIDEO
HDTV PANEL SPLITTER
140
150
OUTPUT COMPRESSED BITSTREAM
VIDEO FIFO
PACKET PROCESSOR
124
COMPRESSOR
125
COMPRESSOR
160
126
COMPRESSOR
MASTER COMPRESSION CONTROLLER
127
COMPRESSOR
165
BUFFER

FIG. 1

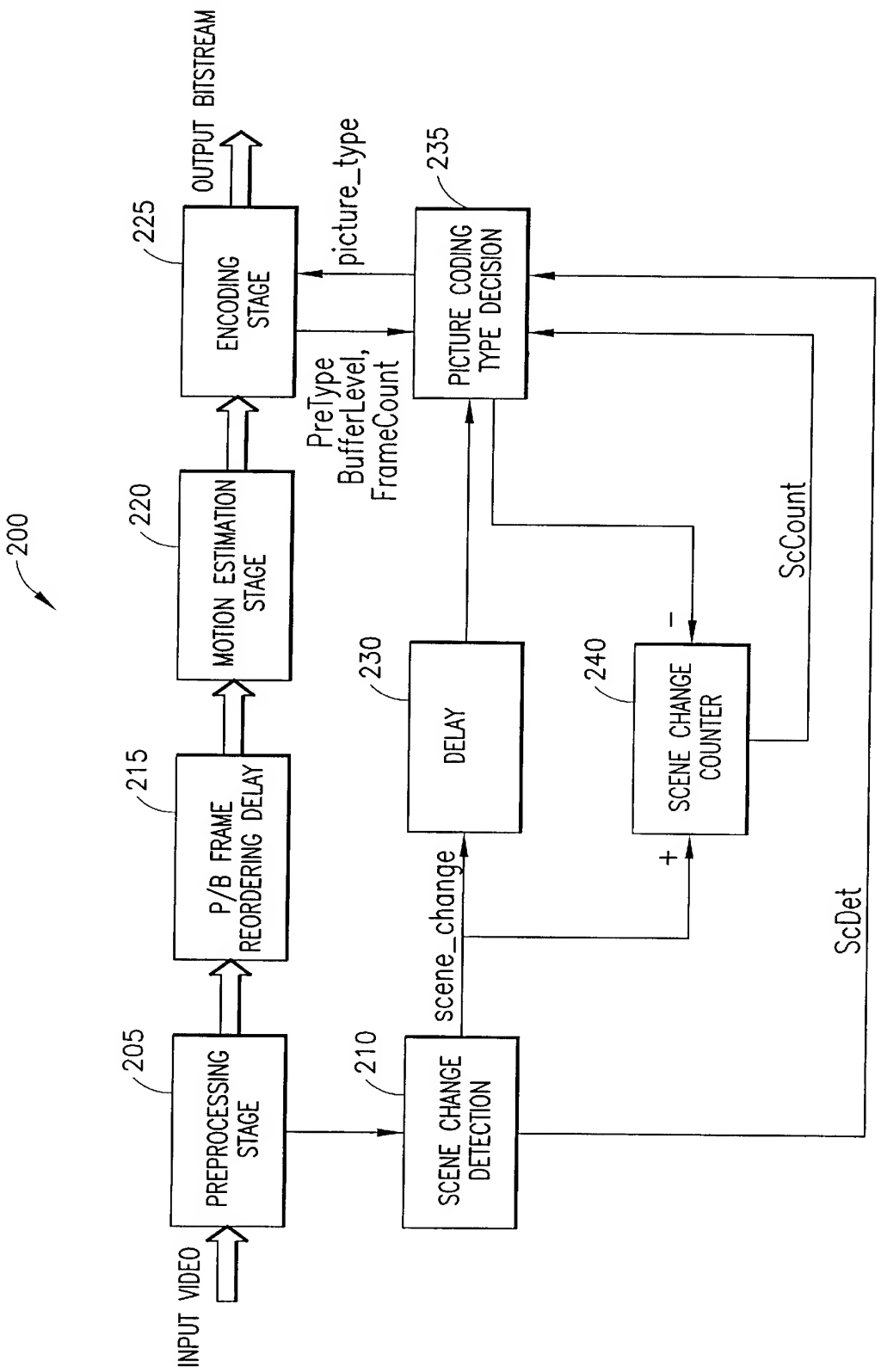

200
INPUT VIDEO
205
PREPROCESSING STAGE
215
P/B FRAME REORDERING DELAY
220
MOTION ESTIMATION STAGE
225
ENCODING STAGE
OUTPUT BITSTREAM
210
SCENE CHANGE DETECTION
scene_change
230
DELAY
PreType
BufferLevel,
FrameCount
235
PICTURE CODING TYPE DECISION
picture_type
240
SCENE CHANGE COUNTER
+
−
ScCount
ScDet

FIG.2

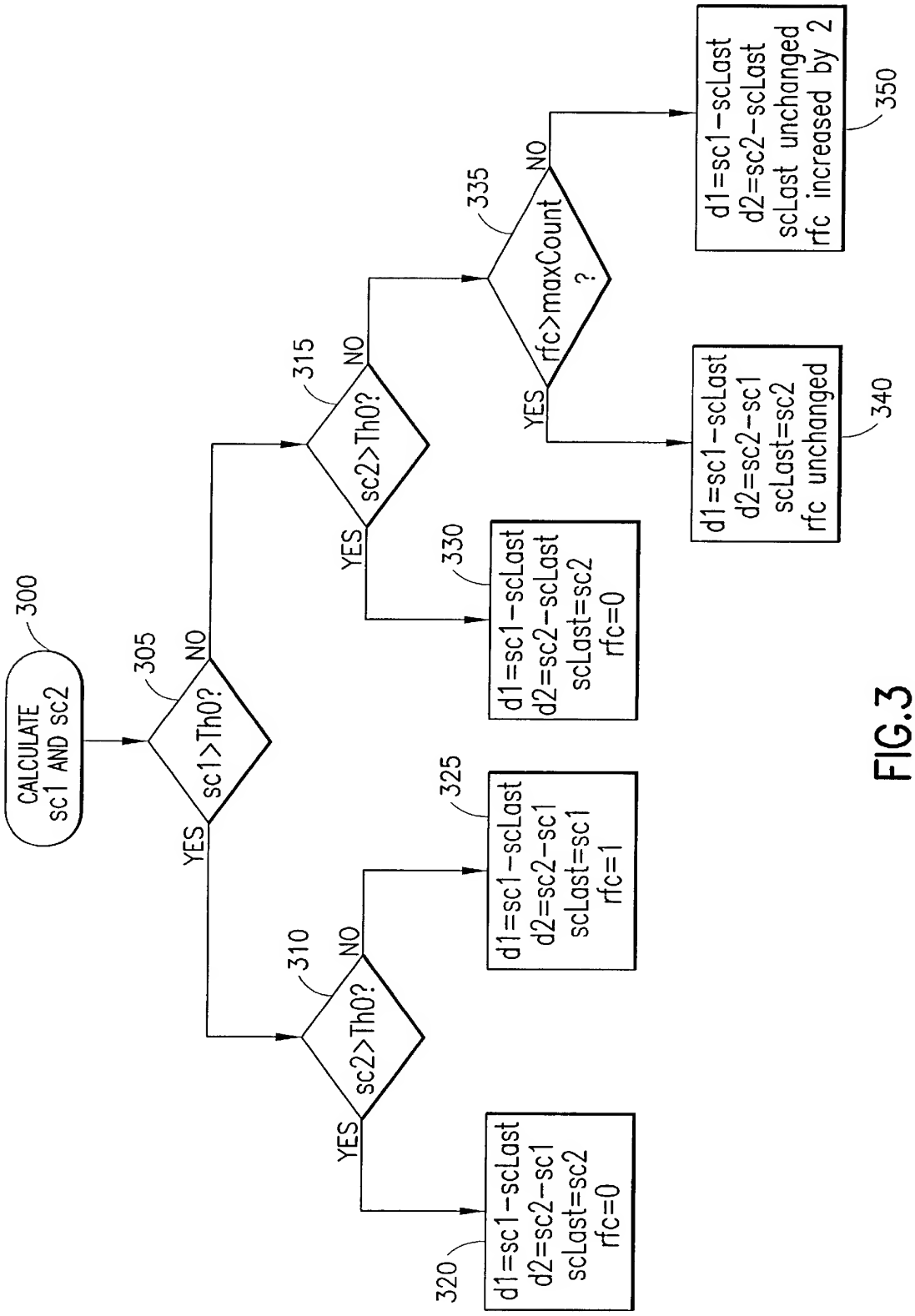

CALCULATE sc1 AND sc2
300
sc1>Th0?
305
NO
YES
sc2>Th0?
310
NO
YES
d1=sc1-scLast d2=sc2-sc1 scLast=sc2 rfc=0
320
d1=sc1-scLast d2=sc2-sc1 scLast=sc1 rfc=1
325
sc2>Th0?
315
NO
YES
d1=sc1-scLast d2=sc2-scLast scLast=sc2 rfc=0
330
rfc>maxCount ?
335
NO
YES
d1=sc1-scLast d2=sc2-sc1 scLast=sc2 rfc unchanged
340
d1=sc1-scLast d2=sc2-scLast scLast unchanged rfc increased by 2
350

FIG.3

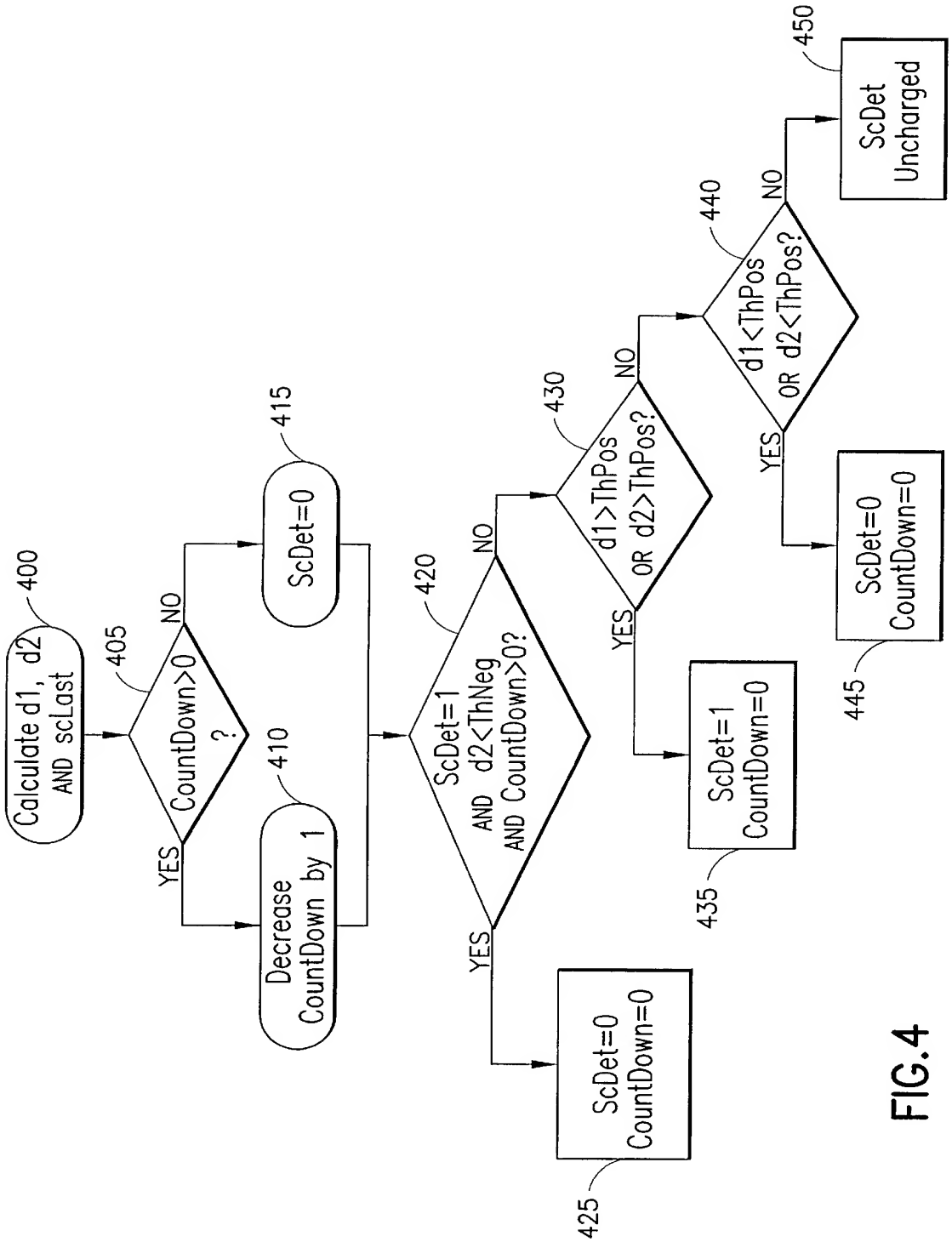

Calculate d1, d2 AND scLast
400
CountDown>0 ?
405
YES
NO
Decrease CountDown by 1
410
ScDet=0
415
ScDet=1 AND d2<ThNeg AND CountDown>0?
420
YES
NO
ScDet=0 CountDown=0
425
d1>ThPos OR d2>ThPos?
430
YES
NO
ScDet=1 CountDown=0
435
d1<ThPos OR d2<ThPos?
440
YES
NO
ScDet=0 CountDown=0
445
ScDet Uncharged
450

FIG.4

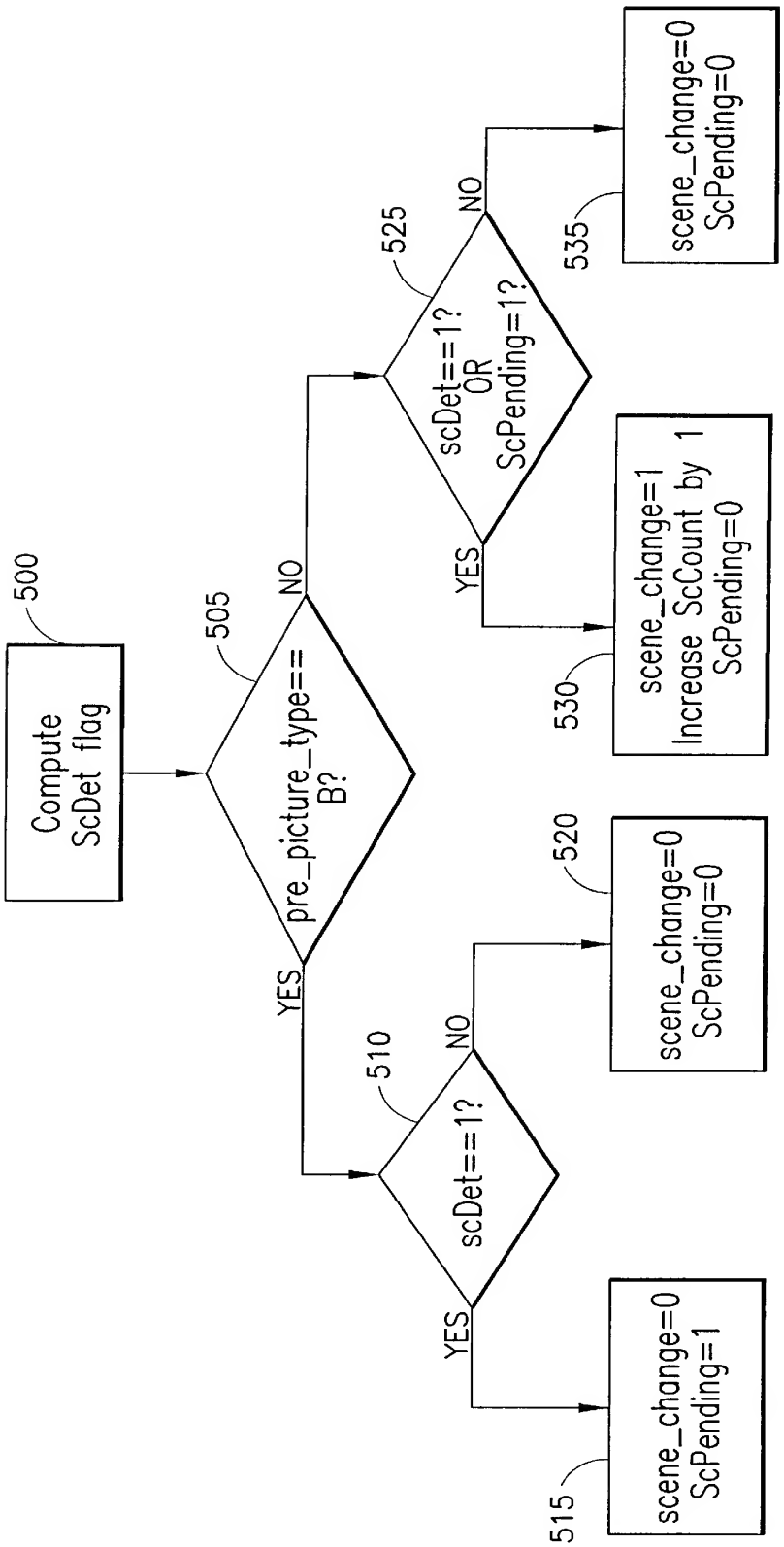

500
Compute ScDet flag
505
pre_picture_type== B?
YES
NO
510
scDet==1?
YES
NO
515
scene_change=0 ScPending=1
520
scene_change=0 ScPending=0
525
scDet==1? OR ScPending=1?
YES
NO
530
scene_change=1 Increase ScCount by 1 ScPending=0
535
scene_change=0 ScPending=0

FIG.5

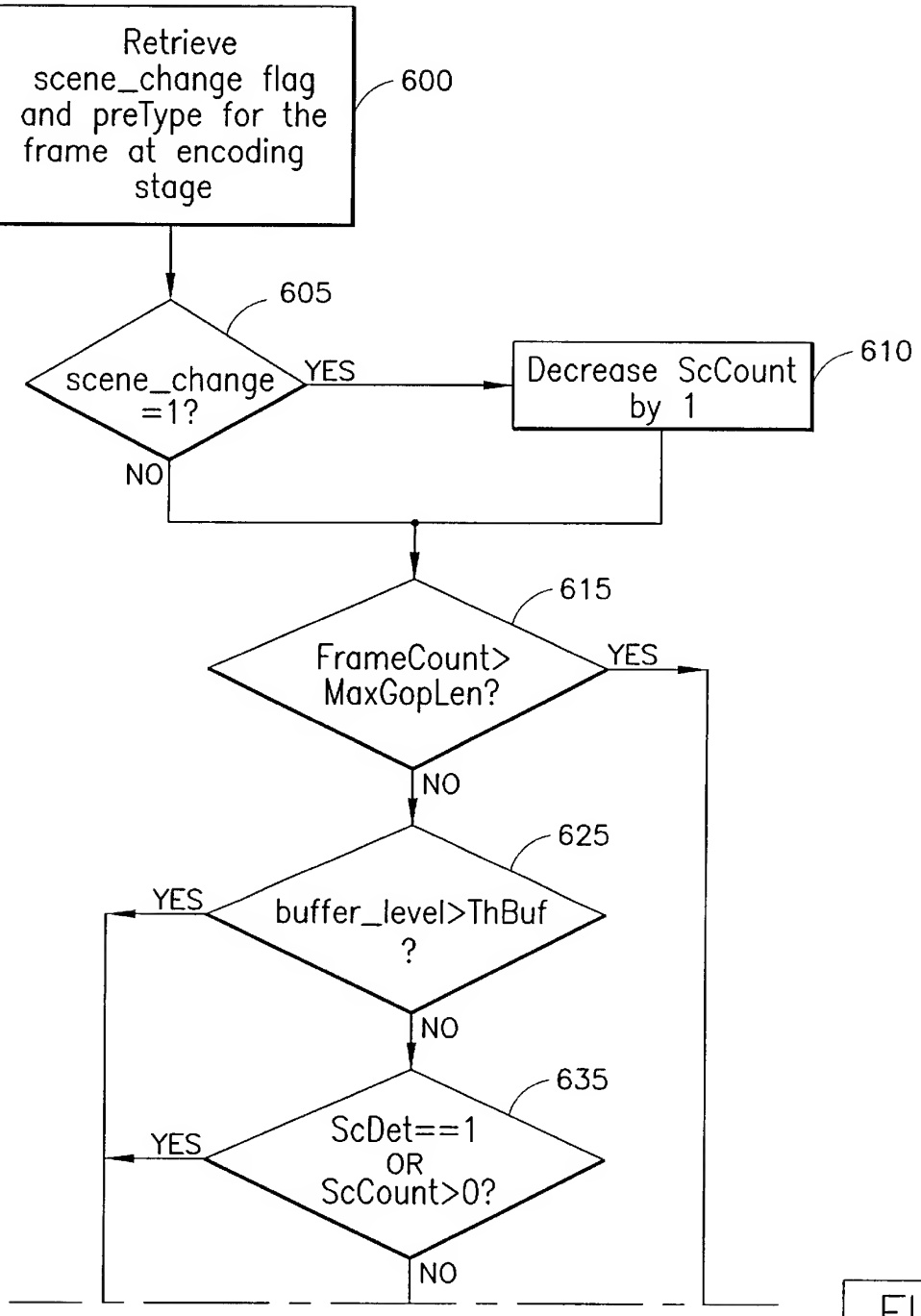


FIG.6A

FIG.6A
FIG.6B

FIG.6

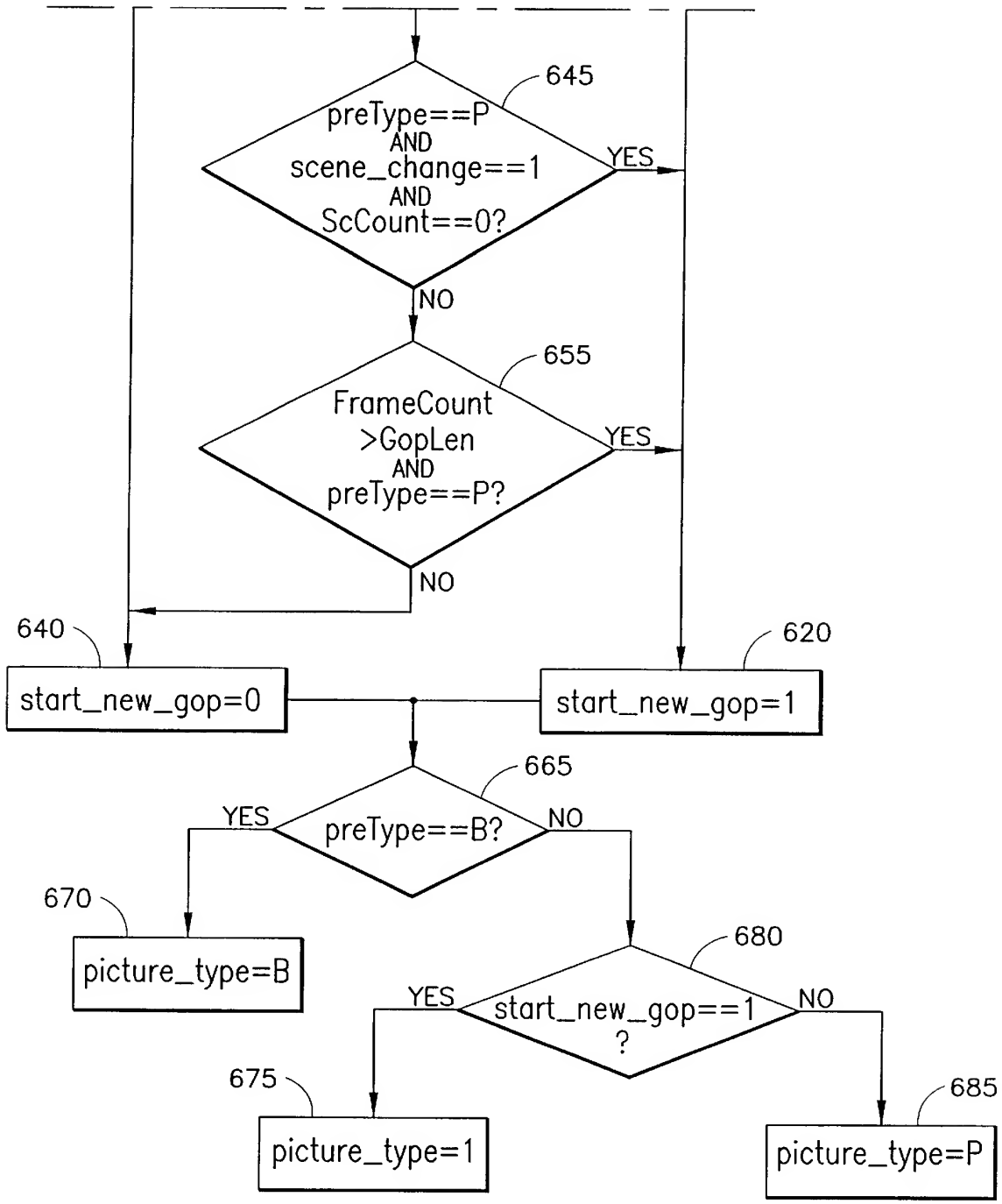

645
preType==P
AND
scene_change==1
AND
ScCount==0?
YES
NO
655
FrameCount
>GopLen
AND
preType==P?
YES
NO
640
start_new_gop=0
620
start_new_gop=1
665
YES
preType==B?
NO
670
picture_type=B
680
YES
start_new_gop==1
?
NO
675
picture_type=1
685
picture_type=P

FIG.6B

METHOD AND APPARATUS FOR DETECTING SCENE CHANGES AND ADJUSTING PICTURE CODING TYPE IN A HIGH DEFINITION TELEVISION ENCODER

This application claims the benefit of U.S. Provisional Application No. 60/102,234, filed Sep. 29, 1998.

BACKGROUND OF THE INVENTION

The present invention relates to video compression, and more particularly to a method and apparatus for detecting scene changes and adjusting the picture coding type to optimize the coding efficiency of a video encoder. The invention is particularly useful for picture coding type decision and scene change detection in a digital HDTV encoder.

Aligning intra-coded (I) frames with scene changes can significantly improve the coding efficiency of an MPEG (Moving Picture Expert's Group) video encoder. In the past, such scene change detection has been provided, for example, by detecting variations in luminance values.

In some existing scene change detection proposals, the coding of an I-frame is avoided until the scene change is over and a new GOP (group of pictures) is started. However, scene change detection is performed on a frame-by-frame basis. Therefore, this approach can yield incorrect results where there are bad edits, special effects or camera flashes, where the scene change might occur on the odd/even field boundary of the same frame.

Accordingly, it would be advantageous to provide a system for detecting scene changes, including flashes, or bad fields that result from improper editing, and adjusting the picture coding type without the aforementioned problems encountered by the prior art. It would be further advantageous to better optimize the coding efficiency of a video encoder during scene changes. It would also be desirable to provide an efficient system for scheduling a new GOP based on the detection of a scene change. Moreover, the system should detect scene changes between successive fields.

The system should make use of an encoding processing pipeline architecture that provides the required lookahead delay to avoid issuing a scheduled I-frame at the proximity of a scene change, while minimizing the amount of frame buffer memory required to provide the lookahead capability.

The scheme should be compatible with different HDTV modes/pixel resolutions, including 1920×1080 I (interlaced scan), 1440×1080 I, and 1280×720 p (progressive scan), as well as standard definition (SDTV) video.

The system should be compatible with any digital video coding scheme, including MPEG-2.

The system should inhibit the encoding of I-frames when an encoder buffer level is too high.

The system should provide a watchdog counter that resets the scene change indication to avoid a perpetual scene change state for transitions from still to motion.

For an MPEG film mode frame picture that is determined to be a scene change frame, to improve coding accuracy and efficiency, the system should deactivate the MPEG-recommended frame-based Discrete Cosine Transform (DCT) and prediction encoding when a scene change may have occurred on a field boundary.

The present invention provides a scene change and adjustment scheme having the aforementioned and other advantages.

SUMMARY OF THE INVENTION

The present invention relates to an efficient video compression scheme that detects scene changes between successive fields, including flashes, or bad fields that result from improper editing, and adjusts the picture coding type and GOP boundaries in response thereto. The term "scene change" is thus used generally herein to encompass events including a normal scene change (at a frame boundary), a scene change at a field boundary of the same frame, a bad edit or flash, or any other sudden change in a sequence of video images.

In accordance with the present invention, scene change detection is performed at a preprocessing stage of a video encoder. The final decision to encode a frame as an I- or P-frame is not made until a final encoding stage. That is, the encoder's processing pipeline is used as a lookahead buffer to minimize the amount of required frame buffer memory.

In particular, the preprocessing stage of the video encoder calculates a change in the sum of pixel differences between consecutive odd fields, or consecutive even fields, for every consecutive input field. A scene change is then detected by looking for a large positive value (exceeding a positive threshold) in the change in sum, followed by a large negative value (less than a negative threshold) I-frames are inhibited at the encoding stage of the processing pipeline as soon as a scene change is detected. A new GOP is started when a scene change frame arrives at the encoding stage of the pipeline, and there is no other scene change frame in the pipeline.

A scene change counter is used to keep track of the number of uncoded scene change frames currently in the processing pipeline. I-frames are prohibited for as long as the scene change counter has a value greater than zero, except for the last scene change frame in a burst (succession) of scene change frames.

Additionally, a, scene change countdown counter, or "watchdog" counter, is used to account for a transition from a still frame to a motion frame to ensure that a scene change is not set permanently.

Moreover, for an MPEG film mode picture that is determined to be a scene change frame, the MPEG-recommended frame-based Discrete Cosine Transform (DCT) and prediction encoding is deactivated when a scene change is indicated at a field boundary (e.g., when there is no redundant field in the picture). Upon such deactivation, either frame- or field-based DCT and prediction can be use on a macroblock-by-macroblock basis in the picture. This allows field prediction to be used to handle bad edits where a scene change may occur at the odd/even field boundary of the same frame.

The invention is suitable for use with both high definition television (HDTV) encoders and standard definition television (SDTV) encoders.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a block diagram illustrating an HDTV encoder in accordance with the present invention.

FIG. **2** illustrates a video encoder processing pipeline in accordance with the present invention.

FIG. **3** illustrates a flowchart for determining scene change score deltas, the most recently used scene change score, and the number of consecutive repeat fields in accordance with the present invention.

FIG. **4** illustrates a flowchart for determining a scene change detected frame, and activating a scene change countdown timer, in accordance with the present invention.

FIG. **5** illustrates a flowchart for setting a scene change flag in accordance with the present invention.

FIG. **6** illustrates a flowchart for determining the final picture coding type for a frame in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to an efficient video compression scheme that detects scene changes between successive fields, and adjusts the picture coding type and GOP boundaries in response thereto.

The following terminology is used:

Term:	Description:
CountDown	Number of remaining frames until ScDet is set to false (far still-to-moving transition sequences);
d1	Delta in consecutive scene change scores for first field;
d2	Delta in consecutive scene change scores for second field;
FrameCount	Number of frames encoded so far in the GOP;
GopLen	Nominal GOP length;
maxCount	Maximum allowed number of consecutive repeat fields;
MaxGopLen	Maximum allowed GOP length;
picture_type	Picture type, possibly modified based on scene change detection;
preType	Pre-picture type - nominal assigned picture type;
rfc	Repeat field count - number of consecutive repeat fields;
sc1	Scene score 1 - scene change metric for first field;
sc2	Scene score 2 - scene change metric for second field;
ScCount	Scene change count - number of scene change frames currently in the processing pipeline;
ScDet	Scene change frame detected;
scene_change	Scene change frame declared;
scLast	Last scene_score;
ScPending	Scene change pending;
start_new_gop	Start a new GOP with current frame;
Th0	Zero threshold for scene scores sc1, sc2;
ThBuf	buffer level threshold that is used to determine whether a new GOP can be started;
ThNeg	Negative threshold for scene scores sc1, sc2; and
ThPos	Positive threshold for scene scores sc1, sc2.

FIG. **1** is a block diagram illustrating an HDTV encoder in accordance: with the present invention.

The encoder **100** includes a HDTV panel splitter **110** that receives an input HDTV video stream. The splitter **110** splits the data among eight separate video compressors **120–127**. The compressors **120** communicate with a Peripheral Component Interconnect (PCI) bus **130**. A video buffer **140** (such as a first-in, first-out, or FIFO buffer) receives the compressed video data from the bus and provides it to a packet processor **150** to provide an output compressed bitstream, A Master Compression Controller (MCC) **160**, which includes a circular buffer **165**, controls the flow of data and synchronizes the video compressors **120–127**. The function of the buffer **165** is discussed further in connection with FIG. **6**.

Note that the invention is illustrated in a multi-compressor HDTV embodiment, but can be used with SDTV data as well. Moreover, the use of eight compressors **120–127** is an example only, as any number of compressors may be used.

The input HDTV picture is split into eight panels. Each panel is processed by a video compression engine. The compression engines **120–127** dump the compressed video data into the video buffer **140**. The Packet Processor **150** pulls the compressed data from the buffer **140** at a rate determined by the configured output bandwidth, and packetizes the data into MPEG transport packets, for example.

FIG. **2** illustrates a video encoder processing pipeline in accordance with the present invention. The video compressors **120–127** process the video frames using a pipeline architecture **200**. The first stage of the pipeline **200** includes a preprocessing stage **205**, which performs video filtering, downsampling (optional), and calculates the statistics for use in a detelecine and scene change detection function **210**.

Subsequent stages of the pipeline **200** include a P/B frame reordering delay function **215** that delays and reorders the video frames. This is done since P-frames are sometimes encoded out of display order for use in predicting the B-frames. A motion estimation stage **220** carries out motion estimation. The final stage includes an encoding stage **225** for performing the actual encoding of the video frames to provide an output bit stream.

To facilitate the subsequent stages of reordering and motion estimation, the preprocessing stage **205** determines whether or not a preprocessed frame is a B-frame. If a frame is not classified as a B frame, the encoder (encoding stage **225**) determines whether it is an I- or P-frame at the final stage of the pipeline, just before the picture is actually encoded. A new GOP is started by an I-frame.

The scene change detection function **210** detects scene changes between consecutive frames at the preprocessing stage of the pipeline. The main objective of scene change processing is to change the location of the scheduled start of a new GOP to align with the start of the new scene if a scene change is detected at the proximity of the originally scheduled I-frame. Once a scene change is detected at the scene change detection function **210**, a control signal (ScDet flag) is sent to the encoding stage **225** to prevent it from generating I-frames for frames that are currently in the encoder's processing pipeline waiting to be encoded. When the corresponding scene change frame subsequently arrives at the encoding stage **225**, it is encoded as an I-frame, assuming there are no other scene change frames in the pipeline, in which case the last scene change frame in the pipeline is encoded as an I-frame.

In particular, the scene change flag is provided to a delay **230** to account for the delays in processing the corresponding frame in the reordering delay function **215** and motion estimation stage **220**, and to a picture coding type decision function **235**. The function **235** sends a corresponding picture_type control signal to the encoding stage **225** to set the picture type of the current frame to be encoded.

In the event that multiple scene changes are detected in a row in successive frames (e.g. camera flashes, special effects, bad frame/field created by improper edit), the start new GOP decision function **235** waits to send the start new GOP control signal to the encoding stage **225**. Thus, the encoding stage **225** waits to encode an I-frame and start a new GOP until the last scene change is over.

This is different from prior art schemes that perform both scene change detection and picture coding type decision at the preprocessing stage, thereby requiring a large amount of lookahead frame buffer, or perform both scene change detection and picture coding type decision at the encoding stage, which does not provide any lookahead capability. Advantageously, the pipeline architecture **200** provides the required lookahead delay to avoid issuing a scheduled I-frame at the proximity of a scene change, while minimizing the amount of frame buffer memory required to provide the lookahead capability.

A scene change counter **240** maintains a count, ScCount, of the number of scene change frames in the pipeline **200**. The count is incremented for each new scene change frame that is designated at the scene change detection function **210**, and decremented for every scene change frame that reaches the picture coding type decision function **235**.

The preliminary picture type information (preType) is determined by the preprocessing stage **205** and tagged to the: video frame throughout the pipeline **200**. This information is retrieved for use in determining the final picture type (picture_type) to be used when the frame arrives at the encoding stage **225**. Essentially, the pre-assigned picture type of a frame may be modified in accordance with the present invention when a scene change is detected, as discussed further in "Scene Change Examples", below.

For every preprocessed frame, the scene change detection function **210** calculates the absolute differences between pixels of the corresponding panel of the current input frame and previous input frame. These absolute differences are summed over the first and second fields and scaled to fit into a 16-bit unsigned integer to form a pair of scene change metrics, namely sc**1** (for the first field) and sc**2** (for the second field). It is possible to generate the scene change metric from a subset of input pixels to simplify implementation, e.g., by spatially down-sampling the frame-prior to calculating the scene change metrics. The scene change detection functions **210** may be implemented by having the MCC **160** read the scene change measures from the video compressors **120–127** to detect a scene change. Based on the results of scene change detection, the encoder's output video FIFO level, and the number of frames coded so far in the GOP, the picture coding type decision function **235** (which may be implemented by the MCC **160**) makes a final decision for the picture coding type of the frame at the encoding stage **225** of the processing pipeline.

FIG. **3** illustrates a flowchart for determining scene change score deltas, the most recently used scene change score, and the number of consecutive repeat fields in accordance with the present invention.

To detect a scene change, the scene change detection function **210**, implemented by the MCC **160**, first calculates the scene change scores sc**1** and sc**2** for the entire odd field and even field of each frame by summing the MAD values over all panels in the frame (block **300**).

The scene change detection function **210** detects a scene change by looking for an abrupt increase in the scene scores followed by an abrupt decrease. At block **305**, Sc**1**>a threshold value Th**0** is tested, and at blocks **310** and **315**, sc**2**>Th**0** is tested. d**1**, d**2**, scLast and rfc are set as indicated at block **320** when both sc**1** and sc**2** exceed Th**0**, or at block **325** when sc**1** exceeds Th**0** but sc**2** does not, or at block **330** when sc**2** exceeds Th**0** but sc**1** does not. At block **335**, a determination is made as to whether the number of consecutive repeat fields (rfc) exceeds the maximum value (maxCount). d**1**, d**2**, scLast and rfc are set accordingly at blocks **340** and **350** if block **335** is true or false, respectively.

In the figures and text herein, "yes", "true" and "1" are synonymous, while "no", "false" and "0" are synonymous.

Generally, there are three thresholds defined {ThPos, ThNeg, Th**0**}, whose values depend on the video format as follows:

	1920 × 1080 I	1440 × 1080 I	1280 × 720 P
ThPos	24,480	17,952	10,240
ThNeg	−16,320	−11,968	−5,120
Th0	6,120	4,488	1,920

The video format refers to the horizontal×vertical pixel resolution, and whether the video is interlaced-scan (I) or progressive-scan (P). This notation should not be confused with the I- and P-frame types. The meaning should be clear from the context. The invention accommodates both interlaced-scan and progressive-scan frames. Progressive scan images are processed in first and second fields. These are suggested thresholds only, and their values may adjusted, e.g., by experimentation with different video sequences.

The scene change detection function **210** calculates the differences, d**1** and d**2**, between consecutive scene change scores for both the first and second input fields, respectively. The most recently used scene change score is maintained by the variable scLast. Repeated fields up to a maximum count (maxCount) are skipped when calculating the differences d**1** and d**2**.

The variable rfc is maintained to keep track of the number of consecutive repeat fields. The maximum number of consecutive repeat fields, maxCount, can be set, e.g., to fourteen fields for 1080 I mode, and twenty-four fields for 720 P mode. However, other values can be used. Whenever rfc exceeds maxCount (box **335**), it is assumed that there is a true still picture so that d**1**, d**2**, and rfc are updated normally (box **340**).

d**1** is calculated as the difference between sc**1** and sc**2**. d**2** is calculated as the difference between sc**2** and either sc**1** or scLast, depending on whether the scene scores are above Th**0** and rfc is smaller than the maximum count.

Small scene change scores for up to the maximum count (maxCount) are excluded (blocks **325**, **330** and **350**) to avoid false detection when there are repeated fields, e.g., in cartoons, slow motion edits, and bad films that fail the detelecine processing. Detelecine processing refers to removing the redundant fields in a 3:2 pulldown video sequence. If the telecine process (transfer from film to video) is noisy, the encoder's detelecine process may not be able to detect the redundant fields.

Another scenario is when the encoder deliberately maintains the redundant field as commanded by the user (e.g., if the user deliberately disables the detelecine process on the encoder), or at edit points where a redundant field may not be dropped (e.g. when there are redundant fields in two consecutive frames). There are also special films which do not follow the regular 3:2 pulldown pattern, e.g., 3:3:2:2 instead of 3:2:3:2.

In the case of a 3-field film frame, we only check for a scene change between the first and second fields, because the third field is dropped at the encoder.

FIG. **4** illustrates a flowchart for determining a scene change detected frame, and activating a scene change countdown timer, in accordance with the present invention.

At block **400**, d**1**, d**2** and scLast are calculated as explained in connection with box **320**, **325**, **330**, **340** or **350** in FIG. **3**.

The MCC maintains the flag ScDet that indicates a scene change has happened. ScDet is set to true (i.e., one) (block **435**) whenever the changes in the scene scores (d**1** or d**2**)

exceed a positive threshold (block **430**). ScDet is reset to zero (blocks **425** and **445**) whenever the changes in d**1** and d**2** are negative, and are less than a negative threshold (block **440**). ScDet is unchanged (block **450**) when d**1** and d**2** are near Th**0**, that is, neither d**1** nor d**2** exceeds ThPos or is less than ThNeg.

A watchdog counter, CountDown, is maintained to keep track of the number of frames elapsed since the last time the scene scores exceed the positive threshold. The counter is decremented by one for every frame thereafter (block **410**) until it reaches zero (block **405**). ScDet is reset to zero when the counter reaches zero (block **415**). This mechanism forces ScDet to reset in case a large negative change in scene score does not follow a large positive surge, which may happen when objects begin to move in a still image (i.e., in a still-to-motion sequence).

FIG. **5** illustrates a flowchart for setting a scene change flag in accordance with the present invention.

ScDet is determined at block **500** as discussed in connection with blocks **415**, **425**, **435**, **445** and **450** in FIG. **4**.

The MCC uses the preliminary picture type (B-frame or not B-frame) (block **505**) determined at the preprocessing stage **205** of the encoding pipeline **200** to indicate that a new GOP may start at the current frame. If a scene change is detected on a pre-processed B-frame (block **510**), Scpending is set to true, but a scene change for the B-frame is not-set (block **515**). If a scene change is not detected on a preprocessed B-frame (block **510**), Scpending and the scene_change flag for the B-frame are set to false (block **520**).

For the next P-frame that follows the pre-processed B-frame, where ScDet or ScPending is true (block **525**), the scene change flag is set to true (block **530**). For the next P-frame that follows the pre-processed B-frame, where neither ScDet nor Scpending is true (block **525**), the scene change flag is set to false (block **535**).

Thus, the MCC keeps track of the number of scene changes currently in the processing pipeline by a ScCount. The value of ScCount increases by one (block **530**) if the scene-change flag is set, and decreases by one (block **610** in FIG. **6**) if the current picture at the encoding stage is a scene change frame. The scene change flag and the preType of the preprocessed frame are stored in the circular buffer **165** of the MCC **160**, and are retrieved later to determine the final picture type when the frame is encoded. The circular buffer **165** is indexed by a temporal reference tag which is sequence number that indicates the input order of each frame. Note that the input to the circular buffer **165** is in display order, while the output is in coding order (after B/P frame reordering).

FIG. **6** illustrates a flowchart for determining the final picture coding type for a frame in accordance with the present invention.

When a frame arrives at the (final) encoding stage of the pipeline, the MCC retrieves the value of the preType and scene_change flag of the frame (block **600**) from its circular buffer. This is performed for the frame to be encoded at the encoding stage **225** of the video compressor. ScCount is decreased by one (block **610**) if scene_change for the current frame is set to true (block **605**). These values are used to determine the final picture coding type (picture type) of the frame in block **670**, **675** or **685**.

A count of the number of frames that has been encoded so far in the GOP (FrameCount) is, maintained. A new GOP is started (block **620**) by converting a P-frame into an I-frame (block **675**) if the FrameCount reaches the user-configured nominal length of a GOP (GopLen) (block **655**), or if a scene change is detected at the current frame and there is no other scene change frame currently in the processing pipeline (as indicated by ScCount=false) (block **645**).

However, even if the above conditions are satisfied, a new GOP is not started (block **640**) if a scene change is detected on any of the uncoded frames in the encoding pipeline (block **635**), or if there is not enough space in the bitstream buffer **140** (FIG. **1**) to accommodate an I-frame (block **625**). An exception is when the FrameCount exceeds the maximum allowable length of a GOP (MaxGopLen) (block **615**), in which case a, new GOP is started regardless (block **620**).

At block **665**, if the preType is a B-frame, there is no change (block **670**). If the preType is a P-frame, it is changed to an I-frame (block **675**) when start_new_gop is true (block **680**). If the preType is a P-frame, it is not changed (block **685**) when start_new_gop is false (block **680**). Note that I- and P-frames are treated the same throughout the pipeline until the encoding stage. It is not necessary to determine whether a frame is an I- or P-frame until the encoding stage.

The start_new_gop flag is used (blocks **620** and **640**) to signal whether or not to start a new GOP with the current frame that is to be encoded. The final value of start_new_gop and picture_type is determined as shown in the flow chart.

The buffer level threshold that is used to determine whether a new GOP can be started or not is calculated as follows:

$$ThBuf = \text{target\_buffer\_level} + (0.5 * GopLen * \text{bit\_rate/frame\_rate}).$$

The target_buffer_level is set to 1/6 of the max_decoder_buffer_level for 1920×1080 I mode, and 1/5 of the max_decoder_buffer_level for 1280×720 P mode. The max_decoder_buffer_level is the maximum number of bits that a decoder buffer that receives the encoded bitstream can hold. Furthermore, the value of ThBuf is capped to not exceed half of the max_decoder_buffer_level.

Scene Change Examples

Tables A–D show four possible scene change scenarios. Other scenarios are possible. In the scenarios, the following notation is used:

X**1**: first field of first scene X

X**2**: second field of first scene X

Y**1**: first field of second scene Y

Y**2**: second field of second scene Y

xx: sumMAD between fields in scene X

xy: sumMAD between fields in scene X and scene Y (assume that xy>>xx)

O: ThNeg<delta<ThPos

+=delta>ThPos

−=delta<ThNeg

For example, in Table A, each frame has first and second fields. For Frames **1** and **2**, X**1** is the first field, and X**2** is the second field. For Frames **3**–**5**, Y**1** is the first field, and Y**2** is the second field. Thus, the boundary between scene X and scene Y is at Frame **3**. Frame **3** is therefore a scene change frame (ScDet=1). The scene change frame is determined by observing the delta value. For Table A, delta transitions from "O" (indicating a small or zero value) to "+" (indicating a large positive value) at the first field (Y**1**) of Frame **3**. Delta then transitions back to "O" at the second field (Y**2**) of Frame **3**, and then to a large negative value (−) at the first

field (Y1) of Frame 4, then back to a small or zero value (O) at the second field of Frame 4.and thereafter.

Note that for the following delta sequence O, O, O, +, O, O, O, . . . ScDet would be set to "1" for the frame with the "+" delta value, and would remain at "1" for each following frames. The watchdog counter (CountDown) described above would cause ScDet to be set back to "0" after ten frames even if delta does not change to "−" to avoid remaining in a perpetual scene change state. This delta sequence could happen when an object begins moving in a still image. On the other hand, the delta sequence O, O, O, −, O, O, . . . results when a motion scene stops and becomes a still picture. In this case, no false scene change will be detected.

As an example of the notation, in Table A, the "xx" notation for Frame 2, field X1, designates the sumMAD between X1 of Frame 1 and X1 of Frame 2. The "xy" notation for Frame 3, field X1, designates the sumMAD between X1 of Frame 2 and X1 of Frame 3. The "yy" notation for Frame 4, field Y1, designates the sumMAD between X1 of Frame 3 and Y1 of Frame 4.

Case 1-A indicates resetting of the picture type based on the scene change detection for a frame sequence B, P, B, P, B. Specifically, in Frame 4, the P-frame is changed to an I-frame. This indicates the start of a new GOP.

Case 1-B indicates resetting of the picture type based on the scene change detection for a frame sequence P, B, P, B, P. Specifically, in Frame 3, the P-frame is changed to an I-frame.

Various other frames sequences are possible.

Note that ScDet is set for an entire frame, even though the detection is based on examining delta for each field.

In Table B, a scene change at the field boundary (between the first field X1 and the second field Y2) of Frame 3. X1 is part of scene X, and Y2 is part of the second scene Y.

Table C illustrates a bad edit, where X, Y and Z denote three separate scenes. A scene change is indicated for Frame 3, where Y1 is the first field of the scene Y, and X2 of Frame 2 is the last field of scene X. No scene change is detected for Frame 4 since the second field of Frame 3 (Z2) and the first field of Frame 4 (Z1) are part of the same scene (z).

Table 4 illustrates a bad edit or flash. The first field of Frame 3 (Y1) is a flash or bad edit scene.

Generally, when a scene change frame is detected, and the scene change frame is a P-frame, it is changed to an I-frame. If the scene change frame is not a P-frame, the first P-frame following the scene change frame is changed to an I-frame.

Note that these cases show an open GOP, where each I- or P-frame is separate by one or more B-frames. However, an open GOP is not required.

TABLE A

Normal Scene Change (at frame boundary):

	Frame 1		Frame 2		Frame 3		Frame 4		Frame 5	
Input fields:	X1	X2	X1	X2	Y1	Y2	Y1	Y2	Y1	Y2
SumMAD:	xx	xx	xx	xx	xy	xy	yy	yy	yy	yy
Delta		O	O	O	+	O	−	O	O	O
ScDet			0		1		0		0	
case 1-A:										
preType	B		P		B		P		B	
picture_type	B		P		B		I		B	
case 2-A:										
preType	P		B		P		B		P	
Picture_type	P		B		I		B		P	

TABLE B

Scene Change at field boundary of the same frame:

	Frame 1		Frame 2		Frame 3		Frame 4		Frame 5	
Input fields:	X1	X2	X1	X2	X1	Y2	Y1	Y2	Y1	Y2
SumMAD:	xx	xx	xx	xx	xx	xy	xy	yy	yy	yy
delta		O	O	O	O	+	O	−	O	O
ScDet			0		1		0		0	
case 1-B:										
preType	B		P		B		P		B	
picture_type	B		P		B		I		B	
case 2-B:										
preType	P		B		P		B		P	
picture_type	P		B		I		B		P	

TABLE C

	Bad edits:											
	Frame 1		Frame 2		Frame 3		Frame 4		Frame 5		Frame 6	
Input fields:	X1	X2	X1	X2	Y1	Z2	Z1	Z2	Z1	Z2	Z1	Z2
SumMAD:	xx	xx	xx	xx	xy	xz	yz	zz	zz	zz	zz	zz
delta		O	O	O	+	+	+	–	O	O	O	O
ScDet			0		1		0		0		0	
case 1-C:												
preType	B		P		B		P		B		P	
picture_type	B		P		B		I		B		P	
case 2-C:												
preType	P		B		P		B		P		P	
picture_type	P		B		I		B		P		B	

TABLE D

	Bad edits or flashes:											
	Frame 1		Frame 2		Frame 3		Frame 4		Frame 5		Frame 6	
Input fields:	X1	X2	X1	X2	Y1	X2	X1	X2	X1	X2	X1	X2
SumMAD:	xx	xx	xx	xx	xy	xx	xy	xx	xx	xx	xx	xx
delta		O	O	O	+	–	+	–	O	O	O	O
ScDet			0		1		0		0		0	
case 1-D:												
preType	B		P		B		P		B		P	
picture_type	B		P		B		I		B		P	
case 2-D:												
Pre Type	P		B		P		B		P		B	
picture_type	P		B		I		B		P		B	

Frame_Pred_Frame_DCT decision:

MPEG-2 encoders use only frame-based prediction and DCT for film mode pictures. This is achieved by setting the flag frame pred_frame_dct=1 in the bit stream syntax. If frame pred_frame_dct=0, either field- or frame-based prediction and DCT can be used on a macroblock-by-macroblock basis for the picture.

Furthermore, MPEG-2 provides a repeat_first_field flag to signal, when set to one, that the current frame belongs to a film and contains a redundant first field such that the frame is composed by three input fields. Specifically, the first field (top or bottom field as identified by the MPEG value top_field_first) is followed by the other field, then the first field is repeated.

The present inventors have determined that, if a film frame contains a redundant first field (repeat_first_field=1), it is not likely that a scene change has occurred at a boundary between the odd and even fields of this frame. It is more likely that there was a clean scene change at a frame boundary. Therefore, frame_pred_frame_dct=1 is set according to the conventional MPEG-2 scheme, and the entire frame is encoded using frame-based DCT and prediction.

However, to handle the special case when a scene change has occurred at the odd-even field boundary of a film mode frame, the frame_pred_frame_dct flag is overridden in accordance with the present invention to allow either frame- or field-based prediction and DCT on a macroblock-by-macroblock basis. This improves coding efficiency, even with the increased overhead of the bits required to designate whether field- or frame-based prediction and DCT is being used for each macroblock, since the correlation of pre-scene change fields will be high, and the correlation of post-scene change fields will be high, but the correlation of a frame; comprising both pre- and post-scene change fields with other frames comprising only pre- or post-scene change fields will be low.

Thus, when there is a bad edit in film mode frame, the encoder is not restricted to use only frame-based prediction and DCT on the transition frame.

Accordingly, if a scene change is detected on a particular frame, the value of the frame pred_frame_dct flag is set to the same value as the repeat_first_field flag. Otherwise, the frame_pred_frame_dct flag is solely determined by the result of telecine detection such that frame_pred_frame_dct is set to true if film is detected, and set to zero if film is not detected.

It should now be appreciated that the present invention provide a system for detecting scene changes on a field-by-field basis, and adjusting the picture coding type to, optimize the coding efficiency of a video encoder. By aligning I-frames with scene changes, the coding efficiency of an MPEG encoder is significantly improved. The invention is advantageous for both HDTV and SDTV signals. Upon detecting a scene change, including flashes, or bad fields that result from improper editing, the picture coding type is adjusted to optimize the coding efficiency of the video encoder. The starting pictures of a new GOP (i.e., an I-frame) is aligned with a scene change to significantly improve the coding efficiency of an MPEG encoder. In a preprocessing stage, the change in the sum of pixel differences between consecutive odd fields, or consecutive even

fields, is calculated for every consecutive input field. A scene change is detected when a large positive value in the change of sum is followed by a large negative value therein. A decision of which picture type to use is not made until a final encoding stage. I-frames can be inhibited when an encoder buffer level is too high.

A watchdog counter resets the scene change indication to avoid a perpetual scene change state for transitions from still to motion.

For an MPEG film mode frame that is determined to be a scene change frame, the MPEG-recommended frame-based Discrete Cosine Transform (DCT) and prediction encoding is deactivated when a scene change may have occurred on a field boundary (e.g., when there is no redundant field in the picture). Upon such deactivation, either frame- or field-based DCT and prediction can be use on a macroblock-by-marcoblock basis in the picture.

Although the invention has been described in connection with various specific embodiments, those skilled in the art will appreciate that numerous adaptations and modifications may be made thereto without departing from the spirit and scope of the invention as set forth in the claims.

For example, while various syntax elements have been discussed herein, note that they are examples only, and any syntax may be used.

What is claimed is:

1. A method for encoding a digital video signal having successive video frames, comprising the steps of:

preprocessing the video signal to calculate, for each successive frame, a first sum of pixel differences between a first field thereof and a first field of a previous frame, and a second sum of pixel differences between a second field thereof and a second field of the previous frame;

calculating, for each successive frame, a change in the first and second sums thereof relative to the first and second sums, respectively, of the previous frame; and

designating a particular one of the successive frames as a scene change frame when the change in at least one of the first or second sums thereof exceeds a positive threshold, and a change in at least one of the first or second sums of a subsequent frame is less than a negative threshold;

wherein the video signal is processed using a pipeline architecture that provides a lookahead buffer capability for encoding the successive video frames.

2. The method of claim **1**, comprising the further step of:

inhibiting the encoding of intra-coded (I) frames when the scene change frame has been designated.

3. The method of claim **1**, comprising the further steps of:

maintaining a count of a number of consecutive scene change frames; and

terminating the scene change designation of the scene change frame for which the count exceeds a maximum value to avoid remaining in a perpetual scene change state for a still-to-motion video transition.

4. The method of claim **1**, comprising the further step of:

maintaining a count of a number of uncoded scene change frames currently in a processing pipeline of a video encoder.

5. The method of claim **4**, comprising the further step of:

inhibiting the encoding of intra-coded (I) frames for as long as the count has a value greater than zero, except for the last scene change frame in a burst of scene change frames.

6. The method of claim **1**, wherein said pipeline architecture includes: (a) a preprocessing stage, wherein said preprocessing, calculating and designating steps occur, and (b) an encoding stage, wherein each frame is encoded, said method comprising the further step of:

changing a pre-assigned picture type of one of the frames at the encoding stage in response to said designating step.

7. The method of claim **6**, wherein:

the pre-assigned picture type for the scene change frame, which is a P-frame, is changed to an I-frame in said changing step.

8. The method of claim **6**, wherein:

the scene change frame is a B-frame, and the pre-assigned picture type for the next successive P-frame is changed to an I-frame in said changing step.

9. The method of claim **1**, comprising the further step of:

commencing the encoding of a new group of pictures (GOP) according to when the scene change frame has been designated.

10. The method of claim **1**, wherein the scene change frame is a film mode frame, comprising the further steps of:

determining whether a scene change boundary between first and second fields of the film mode frame is indicated; and

allowing both frame- and field-based prediction and Discrete Cosine Transform (IDCT) encoding of the film mode frame on a macroblock-by-macroblock basis when said boundary is indicated in said determining step.

11. The method of claim **10**, wherein:

said determining step determines that a scene change boundary between the first and second fields of the film mode frame is indicated when there are no repeated fields in the film mode frame.

12. The method of claim **10**, comprising the further step of:

allowing only frame-based prediction and Discrete Cosine Transform (DCT) encoding of the film mode frame when said determining step determines that a scene change ,boundary between the first and second fields of the film mode frame is not indicated.

13. The method of claim **12**, wherein:

said determining step determines that a scene change boundary between the first and second fields of the film mode frame is not indicated when there is a repeated field in the film mode frame.

14. The method of claim **1**, comprising the further steps of:

monitoring a fullness level of a buffer that receives encoded data of the video signal; and

inhibiting the encoding of intra-coded (I) frames according to when the fullness level exceeds a maximum value.

15. The method of claim **1**, comprising the further step of:

inhibiting the encoding of intra-coded (I) frames when the scene change frame has been designated until the scene change frame is encoded and there are no other scene change frames designated but not yet encoded.

16. A method for encoding a digital video signal having successive video frames, comprising the steps of:

designating a particular one of the successive frames as a scene change frame according to scene change criteria;

wherein the scene change frame is a film mode frame;

determining whether a scene change boundary between first and second fields of the film mode frame is indicated; and

allowing both frame- and field-based prediction and Discrete Cosine Transform (DCT) encoding of the film mode frame on a macroblock-by-macroblock basis when said boundary is indicated in said determining step.

17. The method of claim **16**, wherein:

said determining step determines that a scene change boundary between the first and second fields of the film mode frame is indicated when there are no repeated fields in the film mode frame.

18. The method of claim **16**, comprising the further step of:

allowing only frame-based prediction and Discrete Cosine Transform (DCT) encoding of the film mode frame when said determining step determines that a scene change boundary between the first and second fields of the film mode frame is not indicated.

19. The method of claim **18**, wherein:

said determining step determines that a scene change boundary between the first and second fields of the film mode frame is not indicated when there is a repeated field in the film mode frame.

20. An apparatus for encoding a digital video signal having successive video frames, comprising:

a preprocessor for preprocessing the video signal to calculate, for each successive frame, a first sum of pixel differences between a first field thereof and a first field of a previous frame, and a second sum of pixel differences between a second field thereof and a second field of the previous frame;

means for calculating, for each successive frame, a change in the first and second sums thereof relative to the first and second sums, respectively, of the previous frame; and

a scene change detector for designating a particular one of the successive frames as a scene change frame when the change in at least one of the first or second sums thereof exceeds a positive threshold, and a change in at least one of the first or second sums of a subsequent frame is less than a negative threshold.

21. An apparatus for encoding a digital video signal having successive video frames, comprising:

means for designating a particular one of the successive frames as a scene change frame according to scene change criteria;

wherein the scene change frame is a film mode frame;

a scene change detector for determining whether a scene change boundary between first and second fields of the film mode frame is indicated; and

means for allowing both frame- and field-based prediction and Discrete Cosine Transform (DCT) encoding of the film mode frame on a macroblock-by-marcoblock basis when said scene change detector determines said boundary is indicated.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

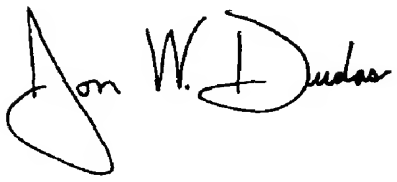
PATENT NO. : 6,731,684 B1
DATED : May 4, 2004
INVENTOR(S) : Wu

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 14,
Line 26, correct "(IDCT)" to read -- (DCT) --
Line 40, delete the comma between the words "change, boundary" so that it reads -- change boundary --

Signed and Sealed this

Thirteenth Day of July, 2004



JON W. DUDAS
Acting Director of the United States Patent and Trademark Office

US 20030193614A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0193614 A1**

Holland et al. (43) **Pub. Date: Oct. 16, 2003**

(54) **METHODS AND APPARATUS FOR CORRECTION OF 2-3 FIELD PATTERNS**

(75) Inventors: **David E. Holland**, Granada Hills, CA (US); **Gavin W. Schutz**, Glendale, CA (US); **Merle C. Sharp**, Los Angeles, CA (US)

Correspondence Address:
O'MELVENY & MEYERS
114 PACIFICA, SUITE 100
IRVINE, CA 92618 (US)

(73) Assignee: **Ascent Media Group, Inc.**, Santa Monica, CA (US)

(21) Appl. No.: **10/430,608**

(22) Filed: **May 5, 2003**

Related U.S. Application Data

(63) Continuation of application No. 09/295,936, filed on Apr. 21, 1999, now Pat. No. 6,559,890.

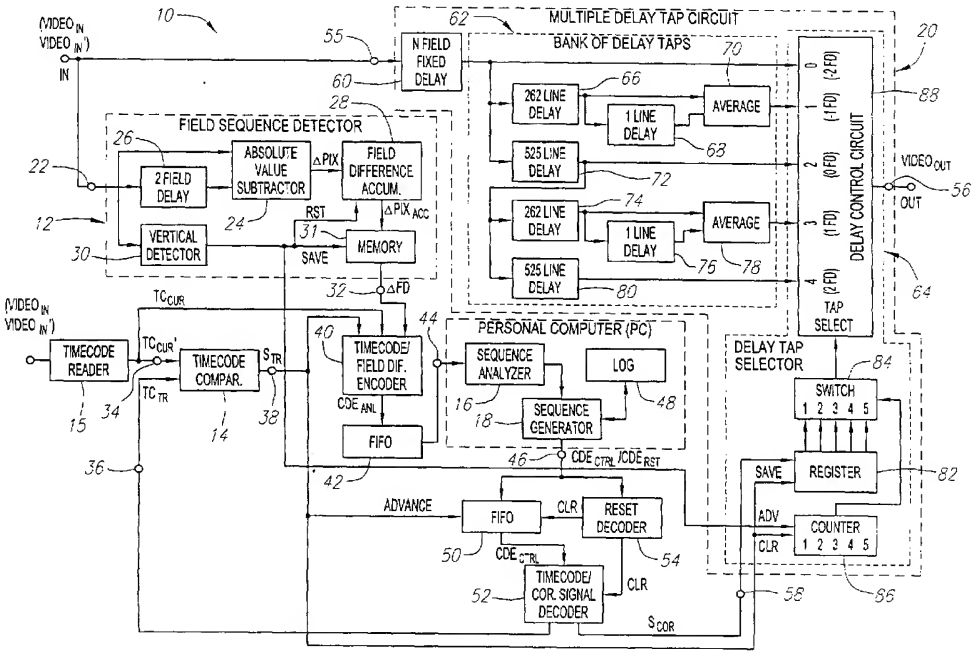
Publication Classification

(51) **Int. Cl.**[7] **H04N 7/01**

(52) **U.S. Cl.** **348/441**; 348/607

(57) **ABSTRACT**

Systems and methods are provided for allowing a user to correct a discontinuous **2-3** field sequence within a disrupted video signal. A **2-3** field pattern fixer can be operated in a one-pass mode and/or a two-pass mode. In the one-pass mode, the disrupted video signal is analyzed to generate correction information, which is used to correct the disrupted video signal as it passes through the 2-3 pattern fixer, resulting in an undisrupted video signal with a continuous **2-3** field sequence. In the two-pass mode, the disrupted video signal is analyzed to generate correction information, which is then stored. This correction information is then used to correct a duplicate of the disrupted video signal, resulting in an undisrupted video signal with a continuous **2-3** field sequence. In this connection, the **2-3** field pattern fixer includes a field sequence detector, a field sequence analyzer, a field sequence generator and a multiple delay tap circuit. The field sequence detector generates field difference values in response to receiving the disrupted video signal. The field sequence analyzer analyzes these field difference values to determine one or more discontinuities within the discontinuous **2-3** field sequence. The field sequence generator generates one or more field sequence correction signals in response to this analysis. The multiple delay tap circuit applies these correction signals to a video signal to generate an undisrupted video signal having a continuous **2-3** field sequence. The **2-3** field pattern fixer can optionally includes a First-In-First-Out (FIFO) memory and a time code comparator, which can be used to store a multitude of the correction signals during the first pass of the two-pass mode, and for synchronizing the application of each of the correction signals to the duplicated disrupted video signal during the second pass of the two-pass mode.



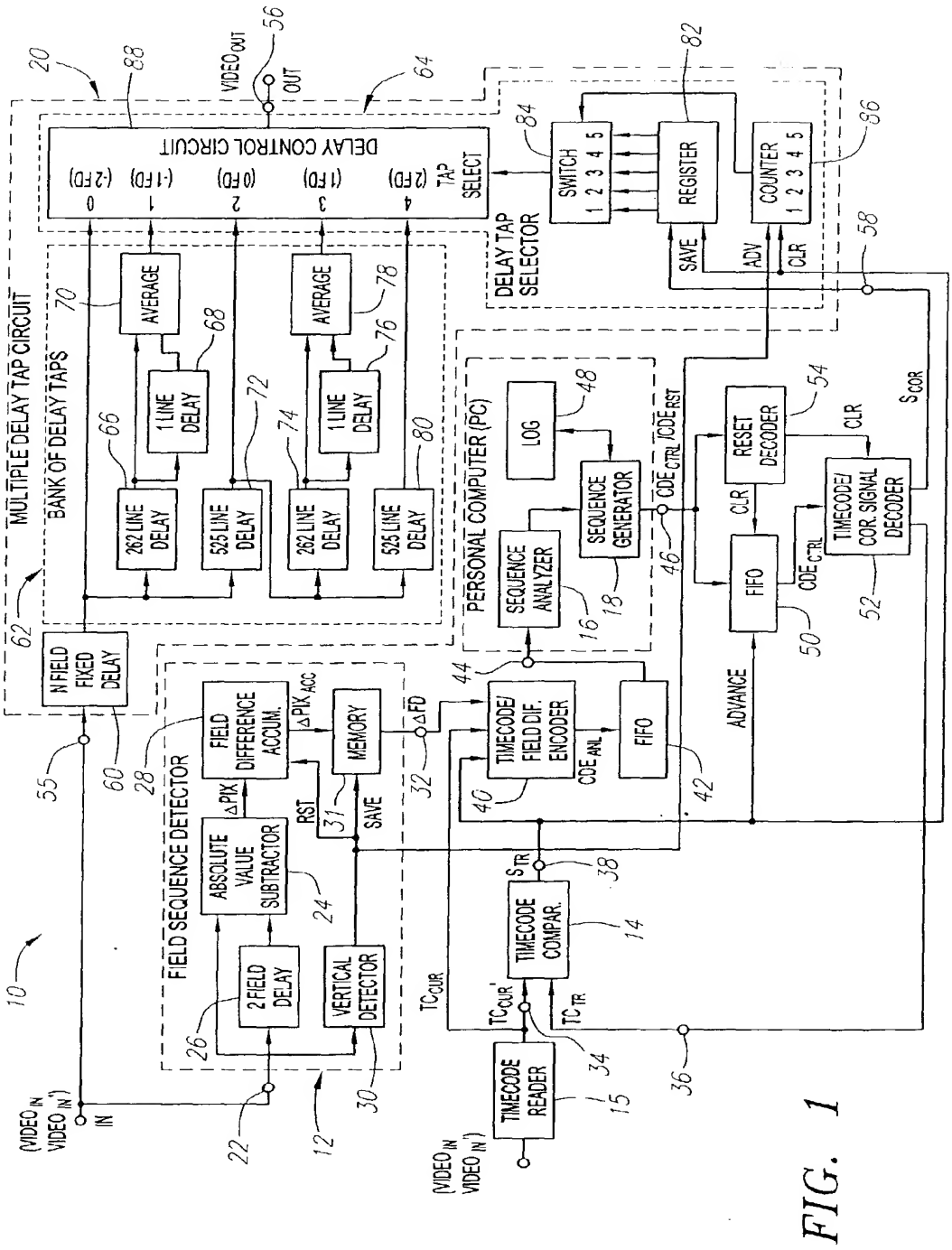
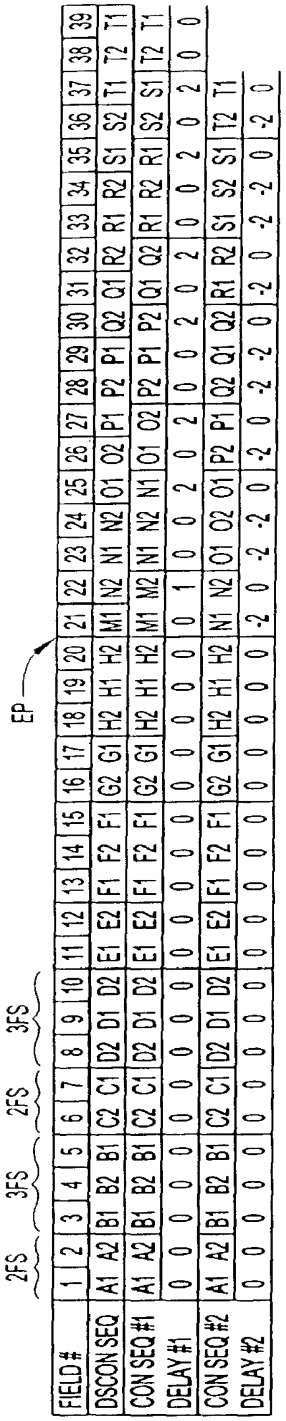

MULTIPLE DELAY TAP CIRCUIT
BANK OF DELAY TAPS
DELAY CONTROL CIRCUIT
DELAY TAP SELECTOR
PERSONAL COMPUTER (PC)
FIELD SEQUENCE DETECTOR
N FIELD FIXED DELAY
262 LINE DELAY
525 LINE DELAY
1 LINE DELAY
AVERAGE
SWITCH
REGISTER
COUNTER
LOG
SEQUENCE GENERATOR
SEQUENCE ANALYZER
RESET DECODER
TIMECODE/ COR. SIGNAL DECODER
FIFO
TIMECODE/ FIELD DIF. ENCODER
FIELD DIFFERENCE ACCUM.
MEMORY
ABSOLUTE VALUE SUBTRACTOR
2 FIELD DELAY
VERTICAL DETECTOR
TIMECODE COMPAR.
TIMECODE READER
ADVANCE
SAVE
ADV
CLR
RST
VIDEO OUT
VIDEO IN

FIG. 1



FIELD #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
DSCON SEQ	A1	A2	B1	B2	B1	C2	C1	D2	D1	D2	E1	E2	F1	F2	F1	G2	G1	H2	H1	H2	M1	N2	N1	N2	O1	O2	P1	P2	P1	Q2	Q1	R2	R1	R2	S1	S2	T1	T2	T1
CON SEQ #1	A1	A2	B1	B2	B1	C2	C1	D2	D1	D2	E1	E2	F1	F2	F1	G2	G1	H2	H1	H2	M1	M2	N1	N2	O1	O2	P1	P2	Q1	P2	Q1	Q2	R1	R2	S1	S2	T1	S1	T2	T1
DELAY #1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	2	0	0	2	0	2	0	0	2	0	2	0	0
CON SEQ #2	A1	A2	B1	B2	B1	C2	C1	D2	D1	D2	E1	E2	F1	F2	F1	G2	G1	H2	H1	H2	N1	N2	O1	O2	O1	P2	P1	Q2	Q1	Q2	R1	R2	S1	S2	S1	T2	T1			
DELAY #2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-2	0	-2	-2	0	-2	0	-2	-2	0	-2	0	-2	-2	0	-2	0		

FIG. 2

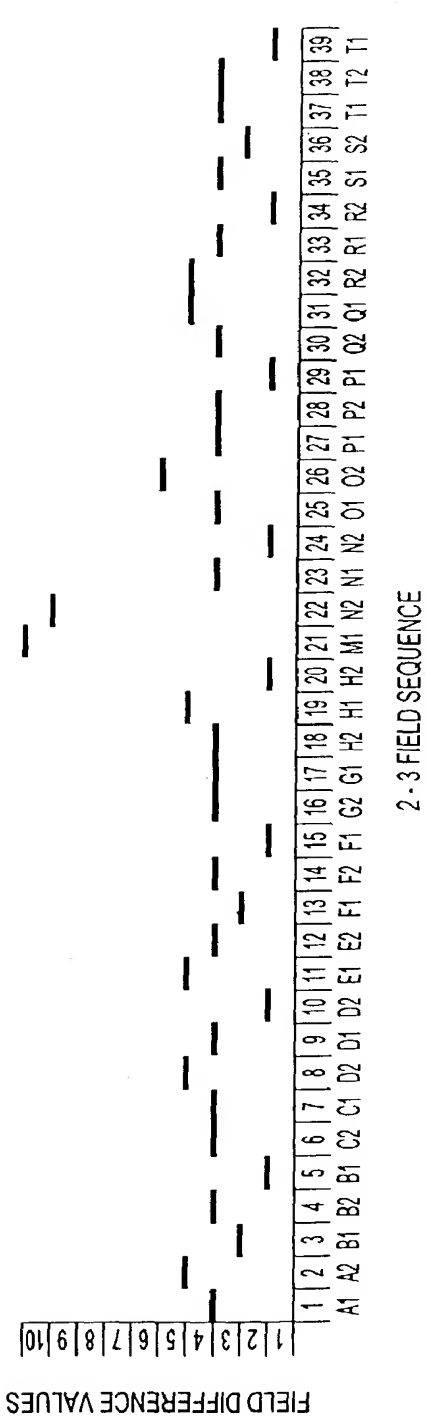


FIG. 3

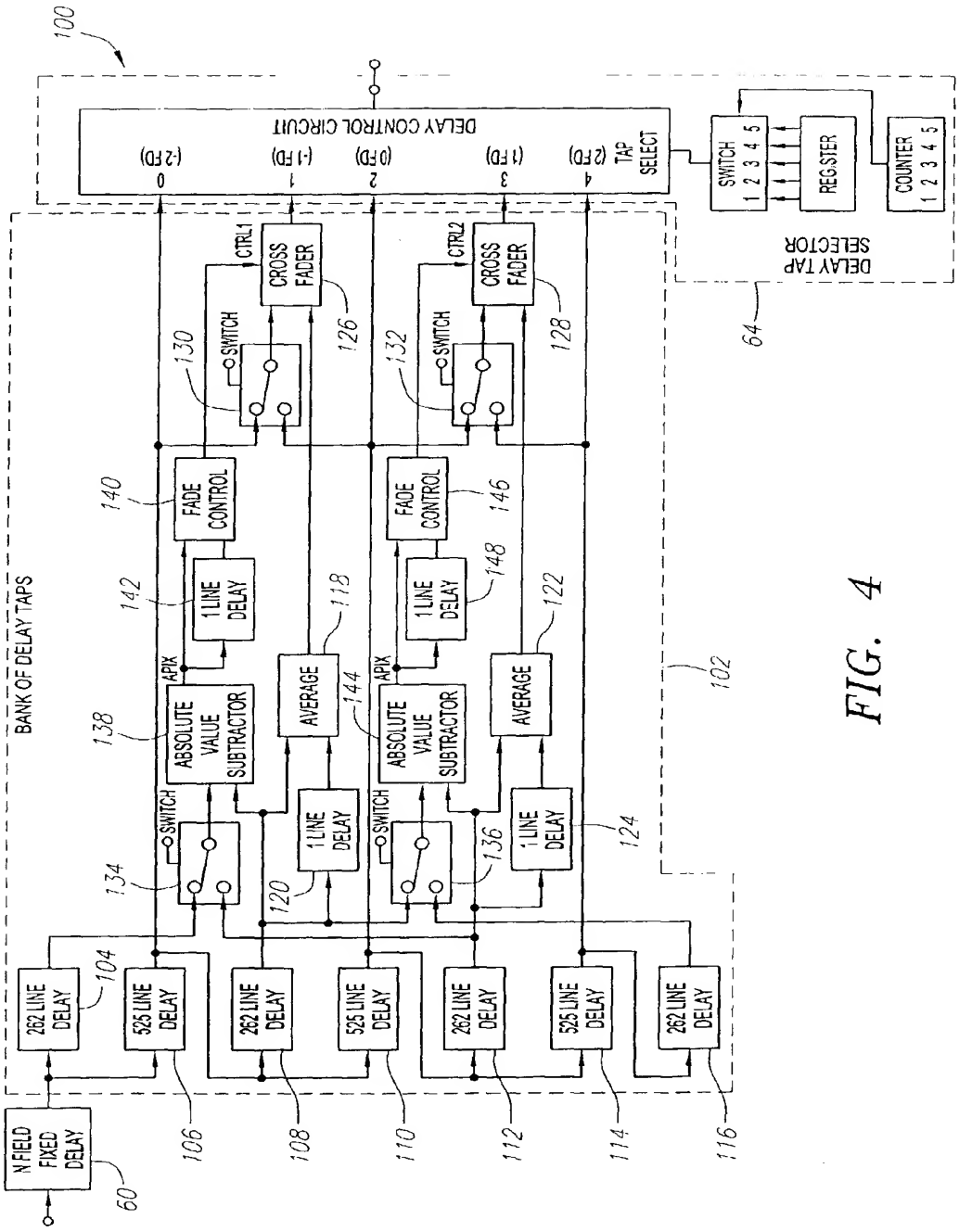

100
DELAY CONTROL CIRCUIT
(-2 FD)
(-1 FD)
(0 FD)
(1 FD)
(2 FD)
TAP SELECT
SWITCH 1 2 3 4 5
REGISTER
COUNTER 1 2 3 4 5
DELAY TAP SELECTOR
64
BANK OF DELAY TAPS
CTRL1
CTRL2
CROSS FADER
126
128
SWITCH
130
132
FADE CONTROL
140
146
1 LINE DELAY
142
148
APIX
ABSOLUTE VALUE SUBTRACTOR
138
144
AVERAGE
118
122
134
136
120
124
102
262 LINE DELAY
104
525 LINE DELAY
N FIELD FIXED DELAY
60
106
108
110
112
114
116

FIG. 4

METHODS AND APPARATUS FOR CORRECTION OF 2-3 FIELD PATTERNS

RELATED APPLICATIONS

[0001] This application is a continuation of Ser. No. **09/295,936**, filed Apr. 21, 1999, now issued as U.S. Pat. No. 6,559,890, incorporated herein by reference.

FIELD OF THE INVENTION

[0002] This invention relates to methods and systems for processing videotape, and more particularly to correcting a discontinuous 2-3 field pattern that resides on the videotape.

BACKGROUND OF THE INVENTION

[0003] In general, telecine machines that operate at 60 field/sec (actually, 59.94 fields/sec) employ a 3:2 pulldown convention to convert film media, which runs at 24 frame/sec (actually, 23.976/sec), to video media, which runs at 60 television field/sec (actually, 59.94 fields/sec). Specifically, a two-field video sequence and a three-field video sequence are alternately generated, with each field sequence corresponding to a film frame. These video fields are interlaced in that the film frames are scanned, such that alternating odd and even fields are generated, with the lines of the odd fields interleaved with the lines of the even fields. For example, a film frame can be scanned to generate a two-field video sequence characterized by an even field and then an odd field (even/odd). The next film frame can be scanned to generate a three-field video sequence characterized by an even field, then an odd field, and then an even field (even/odd/even). The respective first and second even fields in this three-field video sequence are duplicates. The next film frame can be scanned to generate a two-field video sequence characterized by an odd field and then an even field (odd/even). The next film frame can be scanned to generate a three-field video sequence characterized by an odd field, then an even field, and then an odd field (odd/even/odd). The respective first and second odd fields in this three-field video sequence are duplicates. This pattern then repeats for the next four film frames and so on.

[0004] The two/three-field video sequence is sometimes disrupted, such as, e.g., when the video is edited without regard to the video sequence. These disruptions in the video sequence can cause difficulties during processing that requires manipulation of the two/three-field sequence. For example, it is sometimes desired to convert 525 line, 60 fields/sec video to 625 line, 48 fields/sec video. During this procedure, the two/three-field video sequence is converted to a repeating two-field video sequence by removing the duplicate field from each of the three-field video sequences, and, if needed, swapping the order of the two-field video sequence, thereby generating a repeating two-field video sequence characterized by an even field and then an odd field through the entirety of the video (even/odd), or alternatively, an odd field and then an even field through the entirety of the video (odd/even). The 525 line resolution of the 525 line, 60 field/sec video is then interpolated to produce the 625 line, 48 field/sec video with 625 lines of resolution. The resulting 625 line, 48 field/sec video is recorded at 24 frames/sec, which is then played at 25 frames/sec, which is the normal 625 line, 50 field/sec video when viewed. When there is a disruption in the two/three-field video sequence prior to conversion, the two-field sequence subsequent to conversion will sometimes change dominance. That is, the repeating odd/even video sequence changes to a repeating even/odd video sequence. Because a television system cannot process an odd/odd video sequence (or an even/even video sequence), an even field (or odd field) by itself, or an even/odd/even video sequence (or an odd/even/odd video sequence) must be located at the change in dominance. As a result, the video is degraded.

[0005] Another example of a process that requires the manipulation of the two/three-field video sequence is the conversion of video to a digital video disk (DVD). To save memory, the two/three field video sequence is converted to a repeating two-field video sequence in much the same manner described above. The two-field video sequence is then compressed into a motion pictures expert group (MPEG2) format. The DVD player then restores the two/three-field video sequence during playback on the television system. Again, however, during conversion, the field dominance of the two-field video sequence may change, thereby degrading the DVD.

[0006] Thus, it would be desirable to provide methods and systems to correct a disrupted two/three-field video sequence.

SUMMARY OF THE INVENTION

[0007] This present invention comprises novel methods and systems for correcting a discontinuous 2-3 field sequence within a disrupted video signal. A 2-3 pattern fixer constructed in accordance with the present invention can be operated in a one-pass mode and/or a two-pass mode. In a one-pass mode, the disrupted video signal is analyzed to generate correction information, which is used to correct the disrupted video signal as it passes through the 2-3 pattern fixer, preferably in real time, resulting in an undisrupted video signal with a continuous 2-3 field sequence. In a two-pass mode, the disrupted video signal is analyzed to generate correction information, which is then stored. This correction information is then used to correct a duplicate of the disrupted video signal, resulting in an undisrupted video signal with a continuous 2-3 field sequence.

[0008] In a preferred embodiment of the present invention, a 2-3 field pattern fixer includes a field sequence detector, a field sequence analyzer, a field sequence generator and a multiple delay tap circuit. The field sequence detector receives the disrupted video signal and generates a series of field difference values in response thereto by sequentially comparing each of the fields with a field two fields previous. The field sequence analyzer analyzes the series of field difference values and generates field sequence to determine one or more discontinuities within the 2-3 field sequence of the disrupted video signal. The field sequence generator uses this information to generate field sequence reorganization information in the form of a correction signal. In the preferred embodiment, the correction signal comprises a sequence of delays. The correction signal is preferably generated, such that a cumulative delay within the undisrupted video signal is minimized and the number of odd field delays are minimized. The multiple delay tap circuit then applies the correction signal to a video signal to generate an undisrupted video signal having a continuous 2-3 field sequence. That is, selected fields of the disrupted video

signal or duplicate of the disrupted video signal are delayed in accordance with the correction signal, thereby resulting in the undisrupted video signal. The multiple delay tap circuit can optionally include at least one cross-fader to cross-fade between an odd-delayed field and an even-delayed field, thereby minimizing any blur caused by odd field delays.

[0009] This particular embodiment of the 2-3 field pattern fixer can be operated in the one-pass mode. In this connection, the disrupted video signal is analyzed to determine one or more discontinuities with the discontinuous 2-3 field sequence. The discontinuities can be determined by detecting a scene change and a phase change within the 2-3 field sequence. Field sequence reorganization information, and in particular, correction signals are then generated based on these discontinuities. These correction signals are then applied to the disrupted video signal as it passes through the 2-3 field pattern fixer, thereby generating an undisrupted video signal having a continuous 2-3 field sequence from the disrupted video signal.

[0010] The 2-3 field pattern fixer can optionally include a first-in-first-out (FIFO) memory and a time code comparator, allowing the 2-3 field pattern fixer to operate in a two-pass mode. The FIFO is coupled between the field sequence generator and the multiple tap delay circuit and can store several correction signals. The time code comparator is operatively coupled to the FIFO and the multiple delay tap circuit to coordinate the timing of the correction signals as they are input into the multiple delay tap circuit. The time code comparator receives at a first input a current time code of the duplicated disrupted video signal, and at a second input, a trigger time code generated in the field sequence generator and stored in the FIFO. The trigger time code corresponds with the time code during which the next correction signal in the FIFO will be initially applied to the duplicated disrupted video signal. In response thereto, the time code comparator generates a trigger signal that is input into the FIFO and the multiple delay tap circuit.

[0011] In the two-pass mode, the disrupted video signal is analyzed during the first pass to determine one or more discontinuities with the discontinuous 2-3 field sequence. Field sequence reorganization information, and in particular, correction signals and corresponding trigger time codes are then generated based on the discontinuities and associated time codes. These correction signals and corresponding trigger time codes are stored in the FIFO. During the second pass, the duplicated disrupted video signal is received by the multiple tap delay circuit. When the current time code of the duplicated disrupted video signal and the first trigger time code match, the time code comparator sends a trigger signal to the multiple delay tap circuit to begin applying the correction signal to the duplicated disrupted video signal. The trigger signal is also sent to the FIFO to advance the next correction signal for subsequent use by the multiple delay tap circuit, and the next trigger time code to the time code comparator. When the current time code of the duplicated disrupted video signal matches the next trigger time code, the time comparator again sends a trigger signal to the multiple delay tap circuit to begin applying the next correction signal. This process is then repeated until the last correction signal in the FIFO has been applied to the duplicate disrupted video signal. Each of the correction signals are preferably applied to the duplicated disrupted video signal at a field rate equal to the field rate of the disrupted video signal.

[0012] Other and further objects, features, aspects, and advantages of the present invention will become better understood with the following detailed description of the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The drawings illustrate both the design and utility of preferred embodiments of the present invention, in which:

[0014] **FIG. 1** is a block diagram of a preferred embodiment of a 2-3 field pattern fixer constructed in accordance with the present invention;

[0015] **FIG. 2** is table showing the derivation of first and second continuous 2-3 field sequences from a discontinuous **23** field sequence by respectively employing two correction signals;

[0016] **FIG. 3** is a plot of field difference values generated from each of the fields of the discontinuous 2-3 field sequence depicted in **FIG. 2**, wherein the phase and edit point of the discontinuous 2-3 field sequence are particularly indicated by the field difference values; and

[0017] **FIG. 4** is a block diagram of a compensatory multiple delay tap circuit, which can be employed in the **23** field pattern fixer of **FIG. 1**.

DETAILED DESCRIPTION OF THE INVENTION

[0018] **FIG. 1** shows a schematic representation of a 2-3 field pattern fixer **10**, which is configured to receive, at an input IN, a disrupted video signal $VIDEO_{IN}$ having a series of fields arranged in a discontinuous 2-3 field sequence, i.e., a 2-3 field sequence having at least one discontinuity, and transmit, at an output OUT, an undisrupted video signal $VIDEO_{OUT}$ having a series of fields arranged in a continuous 2-3 field sequence, i.e., a 2-3 field sequence having no discontinuities. The 2-3 field pattern fixer **10** is configured to selectively operate in either a one-pass mode or a two-pass mode. In the one pass mode, the 2-3 field pattern fixer **10** analyzes the disrupted video signal $VIDEO_{IN}$ and generates, based on the results of this analysis, the undisrupted video signal $VIDEO_{OUT}$ from the disrupted video signal $VIDEO_{IN}$ as the analysis is being performed. The 2-3 field pattern fixer **10** is configured, such that the disrupted video signal $VIDEO_{IN}$ is corrected in real time. In the two-pass mode, the 2-3 field pattern fixer **10** analyzes several scenes of the disrupted video signal $VIDEO_{IN}$, and stores the results of the analysis (first pass). Based on the stored results of the analysis, the undisrupted video signal $VIDEO_{OUT}$ is then generated from a duplicated video signal $VIDEO_{IN}'$ (second pass). The 2-3 field pattern fixer **10** is configured, such that the disrupted video signal $VIDEO_{IN}$ is corrected in real time during the second pass. In this connection, the 2-3 field pattern fixer **10** generally includes a field sequence detector **12**, a time code comparator **14**, a field sequence analyzer **16**, a field sequence generator **18**, and a multiple delay tap circuit **20**, the arrangement of which will be described in further detail below.

[0019] The field sequence detector **12** detects the discontinuous 2-3 field sequence within the disrupted video signal

$VIDEO_{IN}$, which is serially fed into the input IN, and generates a field difference value ΔFD in response thereto indicative of the detected 2-3 field sequence. In this connection, the field sequence detector **12** includes an input **22** on which the disrupted video signal $VIDEO_{IN}$ is received. The field sequence detector **12** further includes an absolute value subtractor **24**, one input of which is directly coupled to the input **22** and the other input of which is coupled to the input **22** through a two-field delay **26**. Preferably, the two-field delay **26** is implemented using a random access memory (RAM), which can be used to respectively write the serial bits of the video signal to and read the serial bits of the video signal from. It should be noted however, that any circuit that generates a two-field delay can be employed without straying from the principles of this invention. The absolute value subtractor **24** subtracts the pixel values within the current field from the pixel values within a field twice removed in sequence from the current field, and outputs an absolute, unsigned pixel difference value ΔPIX in response thereto.

[0020] The field sequence detector **12** further includes a field accumulator **28**, a vertical detector **30** and a memory device **31**. The field accumulator **28** is coupled to the absolute value subtractor **24** and accumulates the pixel difference values ΔPIX as they are output from the absolute value subtractor **24**. The field accumulator **28** then generates and outputs an accumulated pixel difference value ΔPIX_{ACC}. The memory device **31** is coupled to the field accumulator **28** and stores the accumulated pixel difference value ΔPIX_{ACC} as it is output from the field accumulator **28**. The vertical detector **30** is coupled to the input **22** and detects a vertical synchronization signal within the disrupted video signal $VIDEO_{IN}$, and thus the beginning of the next field. Upon detection of the vertical synchronization signal, the vertical detector **30** simultaneously sends a reset signal RST to the field accumulator **28** to reset the accumulated pixel difference value ΔPIX_{ACC} to 0, and a save signal SAVE to the memory device **31** to save the accumulated pixel difference value ΔPIX_{ACC}, which is then output on an output **32** as the field difference value ΔFD. This process is repeated for each field of the disrupted video signal $VIDEO_{IN}$. As will be discussed in further detail below, the field difference value ΔFD is indicative of the 2-3 field sequence, namely, a phase of the 2-3 field sequence and an edit point, i.e., a scene change, within 2-3 field sequence.

[0021] The time code comparator **14** is employed when the 2-3 field pattern fixer **10** is in the second pass of the two-pass mode, and synchronizes the fields of the duplicated disrupted video signal $VIDEO_{IN}'$ with the corresponding fields of the disrupted video signal $VIDEO_{IN}$, which was previously analyzed during the first pass. A time code reader **15** converts a vertical interval time code (VITC) incorporated within the disrupted video signal $VIDEO_{IN}$ into a current time code TC_{CUR} or a VITC incorporated within the duplicated disrupted video signal $VIDEO_{IN}$ into a current time code TC_{CUR}'. The current time codes TC_{CUR} or TC_{CUR}' are generated in the form of parallel time code data bits. Alternatively, the time code reader **15** can convert a longitudinal time code (LTC), which is sent to the time code reader **15** independently of a video signal, into the current time codes TC_{CUR} or TC_{CUR}'. The time code comparator **14** compares the current time code TC_{CUR}' of the duplicated disrupted video signal $VIDEO_{IN}'$, received at a first input **34**, and a trigger time code TC_{TR} obtained from a previously analyzed disrupted video signal $VIDEO_{IN}$, received at a second input **36**, and outputs a trigger signal S_{TR} at an output **38**. In the preferred embodiment, the trigger signal S_{TR} is a two bit signal. When the current time code TC_{CUR} matches the trigger time code TC_{TR}, the first bit of the trigger signal S_{TR} is high. When the current time code TC_{CUR} is less than the trigger time code TC_{TR}, the first bit of the trigger signal S_{TR} is low. When the current time code TC_{CUR} is greater than the trigger time code TC_{TR}, the second bit of the trigger signal S_{TR} is high, indicating an error condition. The operation of the time code comparator **14** and the significance of the trigger signal S_{TR} will be discussed in further detail below.

[0022] A time code/field difference encoder **40** is coupled to the output **32** of the field sequence detector **12**, the output **38** of the time code comparator **14** and the current time code TC_{CUR}. The time code/field difference encoder **40** encodes the current time code TC_{CUR} (or TC_{CUR}'), trigger signal S_{TR}, and field difference value ΔFD into an encoded character string CDE_{ANL} for use by the field sequence analyzer **16**. An exemplary format for the character string CDE_{ANL} is <AA:BB:CC:DD:E>GGGGGG/CR/LF/. AA:BB:CC:DD:E is encoded from the current time code TC_{CUR}, with AA representing hours, BB representing minutes, CC representing seconds, DD representing video frame number between 0 and 29, and E representing even or odd field. The ">" sign is encoded from the two bits of the trigger signal S_{TR}, and changes to an "=" sign if the current time code TC_{CUR} equals the trigger time code TC_{TR}, or an "?" sign if the current time code TC_{CUR} is greater than the trigger time code TC_{TR}. GGGGGG indicates the field difference value ΔFD. CR represents a carriage return, and LF represents line feed. The character string CDE_{ANL} is input into a First-In-First-Out (FIFO) memory **42**, where several character strings CDE_{ANL} can be stored. The FIFO **42** allows the field sequence analyzer **16** and field sequence generator **18** to utilize the character strings CDE_{ANL} in non-real time. This is especially important if the field sequence analyzer **16** and field sequence generator **18** are embodied in a personal computer (PC), which typically does not read its serial port immediately.

[0023] The field sequence analyzer **16** decodes and analyzes the field difference values ΔFD encoded in the character strings CDE_{ANL}, which are used to determine the 2-3 field sequence of the disrupted video signal $VIDEO_{IN}$. The field sequence generator **18** generates field sequence reorganization information in the form of a field sequence correction signal S_{COR} and a trigger time code TC_{TR}, which are encoded in a character string CDE_{CTRL}. As will be discussed in further detail below, the trigger time code TC_{TR} corresponds to the particular field of the duplicated disrupted video signal $VIDEO_{IN}'$ to which the field sequence correction signal will be applied during the second pass in a two-pass mode. Preferably, the field sequence analyzer **16** and field sequence generator **18** are embodied in a standard personal computer (PC). It should be noted, however, that the field sequence analyzer **16** and field sequence generator **18** can be embodied in logic circuitry without straying from the principles of this invention. The PC includes an RS232 serial input port **44**, which is coupled to the FIFO **42**, and an RS232 serial output port **46**.

[0024] The field sequence analyzer **16** particularly analyzes the field difference values ΔFD to determine the phase

of the discontinuous 2-3 field sequence and any edit point within the 2-3 field sequence. For instance, **FIG. 2** depicts an exemplary discontinuous 2-3 field sequence having alternating distinct two-field sequences 2FS and distinct three field sequences 3FS. The reference letters A, B, C, D, etc., respectively designate the film frames from which the fields are derived. The reference numbers **1** and **2** respectively designate field **1** and field **2** (odd and even, or even and odd). As is apparent, a discontinuity in the form of an edit point EP (or scene change) has been generated between field **H2** and field **M1**, causing, in this case, a change in the phase of the 2-3 field sequence. The field difference values ΔFD output from the field sequence detector **12** are indicative of the first field of a scene change, as well as the third field of a distinct three-field sequence. That is, a comparison of the first two fields immediately subsequent to a scene change with the first two fields previous to each will result in a relatively large field difference value ΔFD for two fields. On the contrary, a comparison of the third and first fields of a distinct three field sequence 3FS will result in a relatively low field difference value ΔFD. If the disrupted video signal $VIDEO_{IN}$ is generated from a telecine digital recording that repeats the third field of the three field frame from digital memory, the field difference value ΔFD may actually be zero. Comparison of a field that is not the first field of a scene change or the third field of a distinct three-field sequence 3FS will result in a nominal field difference value ΔFD providing a reference for determination of the first field of a scene change and the third field of a distinct three-field sequence 3FS.

[**0025**] **FIG. 3** shows a sequence of field difference values ΔFD output from the field sequence detector **12** as each field of the 2-3 field sequence depicted in **FIG. 2** is detected. As can be seen, a high field difference value ΔFD is output from the field sequence detector **12** at field **21** (**M1**) indicating field **M1** as the first field of a scene change. A low field difference value ΔFD is output from the field sequence detector **12** at fields **5** (**B1**), **10**(**D2**), **15**(**F1**), **20**(**H2**), **24**(**N2**), **29**(**P1**), **34**(**R2**), **39**(**T1**), etc., indicating the third field of each of the distinct three-field sequences, which repeats every five fields up until field **24**(**N2**), and then repeats every five fields thereafter.

[**0026**] The field sequence analyzer **16** analyzes the sequence of field difference values ΔFD to determine the first field of the scene change and the change in the phase of the 2-3 field sequence. To determine a phase change within the 2-3 field sequence, the field sequence analyzer **16** correlates the third field of every three-field sequence 3FS to the current time code TC_{CUR}. The field sequence analyzer **16** first converts the current time code TC_{CUR} to an integer, and then performs a modulo **5** operation on this integer to obtain a repeating time code sequence of 0, 1, 2, 3, 4. The field sequence analyzer **16** then correlates the third field of every three-field sequence 3FS with one of these five integers. In a continuous 2-3 field sequence, every third field of a three-field sequence 3FS will correlate with the same integer, indicating a constant phase through the 2-3 field sequence. When there is a discontinuity within the 2-3 field sequence (typically caused by an edit point), the phase of the 2-3 field sequence will usually be altered. For instance, assuming field **1**(**A1**) in **FIG. 2** correlates with the time code integer **0**, the phase of the discontinuous 2-3 field sequence correlates with the time code integer **4** (fields **5**(**B1**), **10**(**D2**), **15**(**F1**), **20**(**H2**)) up until the edit point EP. After the edit point EP, the phase of the discontinuous 2-3 field sequence correlates with the time code integer **3** (fields **24**(**N2**), **29**(**P1**), **34**(**R2**), **39**(**T1**)), thus representing a phase change of −1. An edit point that occurs immediately after the third field of a three-field sequence 3FS or the second field of a two field sequence without a phase change will not generate a discontinuity within the 2-3 field sequence. This condition generally occurs 8% of the time. A phase change in the 2-3 field sequence not resulting from an edit point EP indicates an error condition.

[**0027**] Based on the determination of the first field of the scene change and the phase change of the discontinuous 2-3 field sequence, the field sequence generator **18** generates field sequence reorganization information, which can subsequently be applied to either the disrupted video signal $VIDEO_{IN}$ (one-pass mode) or the duplicated disrupted video signal $VIDEO_{IN}$' (two-pass mode), to generate the undisrupted video signal $VIDEO_{OUT}$. This reorganization information is based on a reorganization of the disrupted 2-3 field sequence internally performed within the field sequence generator **18**. In particular, with knowledge of the first field of the scene change and the phase change of the discontinuous 2-3 field sequence obtained from the field sequence analyzer **16**, the field sequence generator **18** deletes, repeats and/or swaps fields within the discontinuous 2-3 field sequence to generate the continuous 2-3 field sequence. For instance, a distinct two-field sequence 2FS can be changed to a distinct three-field sequence 3FS by repeating one of the fields. A distinct three-field sequence 3FS can be changed to a distinct two-field sequence 2FS by deleting the first or third field of the distinct three-field sequence 3FS. A distinct two-field sequence 2FS composed of an odd field and then an even field can be changed to a distinct two-field sequence 2FS composed of an even field and then an odd field by swapping fields, and vice versa. A distinct three-field sequence 3FS composed of an even field, then an odd field, and then an even field can be changed to a distinct three-field sequence 3FS composed of an odd field, then an even field, and then an odd field by repeating and deleting fields.

[**0028**] As will be described in further detail below, the field sequence generator **18** performs this reorganization by associating a relative delay for certain of the fields of the discontinuous 2-3 field sequence. This delay information is used to generate a delay sequence for each of the fields of the continuous 2-3 field sequence, beginning with the first field of the scene and ending with the last field of the scene (the scene change location may move a field or two after this process). For instance, the discontinuous 2-3 field sequence of **FIG. 2** is shown reconstructed into either a first continuous 2-3 field sequence or a second continuous 2-3 field sequence by assuming a relative delay for certain of the fields of the discontinuous 2-3 field sequence. As is apparent, frame **21**(**M1**) is the first field of the scene change, and the scene change has caused a phase change in the 2-3 field sequence of −1 (i.e., frame **24**(**N2**) is four frames removed from frame **20**(**D2**) instead of five). Based on this, the field sequence generator **18** can determine for certain fields, the delay required to generate a continuous 2-3 field sequence.

[**0029**] For instance, to construct the first continuous 2-3 field sequence, the field sequence generator **18** derives the partial field sequence **M1**/**M2**/**N1**/**N2**/**N1** from the discontinuous 2-3 field sequence by associating a 0 field delay and a 1 field delay with field **21**(**M1**), a 0 field delay and a 2 field

delay with field **23**(N1), and a 0 field delay with field **24**(N2). The field sequence M1/M2/N1/N2/N1 can then be respectively used for fields **21-25**. It should be noted that only the fields of the discontinuous 2-3 field sequence required to construct a continuous 2-3 field sequence have a delay associated therewith. For example, a delay is not associated with field **22**(N2), which is effectively ignored. Carrying this process throughout the scene will result in a delay sequence of 0, 1, 0, 0, 2, 0, 2, 0, 0, 2, 0, 2 for fields **21-32**. As can be seen, a repeating delay sequence of 0,0,2,0,2 occurs after the 1 field delay. Although not depicted, odd field delays can occur at the end of the scene, just like at the beginning of the scene.

[**0030**] To construct the second 2-3 field sequence, the field sequence generator **18** derives the sequence N1/N2/O1/O2/O1 from the discontinuous 2-3 field sequence by associating a −2 field delay with field **23**(N1), a 0 field delay with field **22**(N2), a −2 field delay and a 0 field delay with field **25**(O1), and a −2 field delay for field **26**(O2). A delay is not associated with frame **21**(M1) and frame **24**(N2), which are effectively ignored. Thus, a delay sequence having a repeating delay sequence of −2,0,−2,−2,0, is generated for fields **21-25**, and so on. It should be noted that, in reality, a negative delay cannot be achieved. The negative delays represented in **FIG. 2**, however, are relative to a fixed delay. For example, if there is a fixed delay of 10 fields, a −2 field delay would actually be an absolute **8** field delay. As will be discussed in further detail below, the field sequence generator **18** generates a delay sequence based on various criteria.

[**0031**] The delay sequence is used to generate a correction signal S_{COR}. When the 2-3 pattern fixer **10** is in the two-pass mode, the field sequence correction signal S_{COR} along with the corresponding time code, i.e., the trigger time code TC_{TR}, is stored in a log **48** during the first pass of the two-pass mode. For example, the delay sequence 0, 1, 0,0,2,0,2, 0,0,2,0,2, etc. depicted in **FIG. 2** with respect to the first continuous 2-3 field sequence will result in a correction signal of 0, 1 and a correction signal of 0, 0, 2, 0, 2. The repeating delay sequence of −2,0,−2,−2,0 depicted in **FIG. 2** with respect to the second continuous 2-3 field sequence will result in a single correction signal of −2, 0, −2, −2, 0. A multitude of correction signals S_{COR} and corresponding trigger time codes TC_{TR} are accumulated over several scenes and stored in the log **48**.

[**0032**] The field sequence correction signals S_{COR} and corresponding trigger time codes TC_{TR} are encoded into control character strings CDE_{CTRL} and output on the RS232 serial output port **46**. An exemplary format for the character string CDE_{CTRL} is {AA:BB:CC:DD:E}HHHHH/CR/LF/, where AA:BB:CC:DD:E is the trigger time code TC_{TR} with AA representing hours, BB representing minutes, CC representing seconds, DD representing video frame number between 0 and 29, and E representing even or odd field; HHHHH represents the field sequence correction signal S_{COR}; CR represents carriage return; and LF represents line feed. The character string CDE_{CTRL} is input into a First-In-First-Out (FIFO) memory **50**, where several character strings CDE_{CTRL} can be stored. The FIFO **50** obviates the need for the field sequence analyzer **16** and field sequence generator **18** to respond to each field, eliminating the need of the field sequence analyzer **16** and field sequence generator **18**, i.e., the PC, to operate in real time. The output of the time code comparator **14** is coupled to and inputs the trigger signal S_{TR} into the FIFO **50**. When the 2-3 field pattern fixer **10** is performing the second pass of the two-pass mode, and when the trigger signal S_{TR} indicates that the current time code TC_{CUR} is equal to the trigger time code TC_{TR} as discussed above, the FIFO **50** outputs the next control character string CDE_{CTRL}. When the 2-3 field pattern fixer **10** is in the one-pass mode, the field sequence generator **18** provides the signal to output the control character string CDE_{CTRL}.

[**0033**] A time code/correction signal decoder **52** is coupled to the FIFO **50** and receives the next control character string CDE_{CTRL}. The time code/correction signal decoder **52** outputs, on a first output, the decoded correction signal S_{COR} for subsequent use by the multiple delay tap circuit **20** and on a second output, the trigger time code TC_{TR} for input into the second input **36** of the time code comparator **14**. As will be described in further detail below, application of the trigger time code TC_{TR} to the second input **36** of the time code comparator **36** and subsequent generation of the trigger signal S_{TR} allows the proper correction signal S_{COR} to be applied to the duplicated disrupted video signal $VIDEO_{IN}$' when the 2-3 pattern fixer **10** is in the two-pass mode.

[**0034**] The field sequence analyzer **16** generates a reset code CDE_{RST} if the current time code TC_{CUR} backs up, i.e., the videotape is rewound. The reset code CDE_{RST} is output when the current time code TC_{CUR} either stops or is forwarded, i.e., the videotape is cued or played. The reset decoder **54** is coupled to the field sequence analyzer **16** and decodes the reset code CDE_{RST} upon reset thereof. The output of the reset decoder **54** is coupled to and clears the FIFO **50** and time code/correction signal decoder **52** in response to the reset code CDE_{RST}. The reset code CDE_{RST} allows the field sequence analyzer **16** to clear the control character strings CDE_{CTRL} from the FIFO **50** and input the proper control character strings CDE_{CTRL} when the cleared control character strings CDE_{CTRL} are out of synchronization with the current time code TC_{CUR}.

[**0035**] The multiple delay tap circuit **20** includes a first input **55** to receive the disrupted video signal $VIDEO_{IN}$ (or duplicated video signal $VIDEO_{IN}$') and an output **56** to output the undisrupted video signal $VIDEO_{OUT}$. A second input **58** of the multiple delay tap circuit **20** is coupled to the output of the time code/correction signal decoder **52** to receive the field sequence correction signal S_{COR}. The multiple delay tap circuit **20** includes a fixed delay **60**, a bank of delay taps **62** and a delay tap selector **64**. For the purposes of illustration, the bank of delay taps **62** is shown to include 0, 1, 2, 3, and 4 field delay taps. Each of the fields of the disrupted video signal $VIDEO_{IN}$ or duplicated video signal $VIDEO_{IN}$' can be relatively delayed within a range of between −2 and 2 fields, as discussed above. Preferably, the fixed delay **60** is several fields in length to provide the field sequence analyzer **16** enough time to analyze the discontinuous 2-3 field sequence of the disrupted video signal $VIDEO_{IN}$ and the field sequence generator **18** enough time to generate the field sequence correction signals S_{COR} in response thereto, when the 2-3 field pattern fixer **10** is in the one-pass mode. Preferably, the fixed delay **60** and delay taps **62** are implemented using a random access memory (RAM). It should be noted that the audio signal (not shown) should be delayed the same number of fields equal to the fixed delay **60** plus the delay of the center tap to facilitate the synchro-

nization of the audio signal and the undisrupted video signal $VIDEO_{OUT}$ when applied to videotape.

[0036] The bank of field delay taps **62** include delay taps 0, 1, 2, 3, and 4, which represent alternating even and odd fields of delay, i.e., relative delays of −2, −1, 0, 1, and 2 fields. Delay tap **0** (relative delay of −2 fields) is generated by connecting a direct line between the fixed delay **60** and the delay control circuit **88**. Delay tap **1** (relative delay of −1 field) is generated by connecting a 262½-line delay between the fixed delay **60** and the delay control circuit **88**. The 262½-line delay is created by averaging, through an averaging circuit **70**, a 262 line delay **66** with the 262 line delay **66** in series with a 1-line delay **68**. Delay tap **2** (relative delay of 0 fields) is generated by connecting a 525-line delay **72** between the fixed delay **60** and the delay control circuit **88**. Delay tap **3** (relative delay of 1 field) is generated by connecting the 525 line delay **72** and 262½ line delay between the fixed delay **60** and the delay control circuit **88**. The 262½-line delay is created by averaging, through an averaging circuit **78**, a 262 line delay **74** with the 262-line delay **74** in series with a 1-line delay **76**. Delay tap **4** (relative delay of 2 fields) is generated by connecting the 525-line delay **72** and a 525-line delay **80** between the fixed delay **60** and the delay control circuit **88**. It should be noted that by adjusting the number of lines in each delay appropriately, the 2-3 pattern fixer **10** can be used for other applications, such as, e.g., high definition television (HDTV).

[0037] The delay tap selector **64** includes a register **82**, a switch **84**, a counter **86** and a delay control circuit **88**. The register **82** is connected to the second input **58** and stores the field sequence correction signal S_{COR} received on the second input **58** from the time code/correction signal decoder **52**. The delay numbers within the field sequence correction signal S_{COR} are respectively output onto five lines leading to five takeoffs on the switch **84**. The counter **86**, which continuously counts from 1 to 5, is operatively coupled to the switch **84**, such that the switch **84** serially outputs the delay numbers of the field sequence correction signal S_{COR}. The register **82** and counter **86** each includes an input coupled to the output **38** of the time code comparator **14** to receive the trigger signal S_{TR}. When the trigger signal S_{TR} indicates that the current time code TC_{CUR} equals the trigger time code TC_{TR}, the register **82** saves the field sequence correction signal S_{COR} output from the time code/correction signal decoder **52** and the counter **86** is reset to 1. The counter **86** is keyed to the beginning of each field. In this regard, the output of the vertical detector **30** is coupled to the counter and sends an advance signal ADV thereto at the beginning of each field. In this manner, a delay number is output from the switch **84** at the beginning of each field. The delay control circuit **88** is coupled between the switch **84** and the bank of delay taps **62**. The delay control circuit **88** selects a particular tap corresponding to the delay number received by the delay control circuit **88**. For instance, if the delay number received by the delay control circuit **88** is −2, the delay control circuit **88** will select delay tap **0**. By selecting the appropriate delay taps, the disrupted video signal $VIDEO_{IN}$ (or duplicated disrupted video signal $VIDEO_{IN}'$) received at the first input **55** of the multiple delay tap circuit **20** can be reconstructed to generate the undisrupted video signal $VIDEO_{OUT}$ at the output **56** of the multiple delay tap circuit **20** in accordance with the field sequence correction signal S_{COR} generated by the field sequence generator **18**. It should be noted that the bank of delay taps **62** can include more delay taps than those depicted in **FIG. 1**, providing the field sequence generator **18** increased flexibility in selecting a delay sequence.

[0038] It should be noted that a vertically blurred field is generated when the odd field delay taps (1-field and 3-field delay taps) are used due to the averaging of signals offset from each other by a 1 line delay. As such, it is generally undesirable to generate a delay sequence with an odd number. This undesirable effect, however, will only occur in a minimum number of fields, generally at the beginning and/or the end of a scene.

[0039] The resultant blurred field may be minimized or eliminated by deriving at least a portion of the field from a field that is delayed an even number of fields, by employing the multiple delay tap circuit **100** depicted in **FIG. 4**, instead of the multiple delay tap circuit **20** depicted in **FIG. 1**. The multiple delay tap circuit **100** includes a compensating bank of delay taps **102**, along with the fixed delay **60** and the delay tap selector **64**. As with the bank of delay taps **62**, delay taps **0**, **2** and **4** represent even fields of delay (i.e., relative delays of −2, 0 and 2 fields). Delay tap **0** (relative delay of −2 fields) is generated by connecting a 525-line delay **106** between the fixed delay **60** and the delay control circuit **88**. Delay tap **2** (relative delay of 0 fields) is generated by connecting the 525-line delay **106** and a 525-line delay **110** between the fixed delay **60** and the delay control circuit **88**. Delay tap **4** is generated by connecting the 525-line delays **106** and **110** and a 525-line delay **114** between the fixed delay **60** and the delay control circuit **88**.

[0040] The bank of delay taps **102** are compensating in that delay tap **1** can be cross-faded between a first odd-delayed field, i.e., field with a relative delay of −1 field (−1FD), and one of the two even-delayed fields immediately adjacent the first odd-delayed field, i.e., a field with a relative delay of −2 fields or 0 fields (2FD or OFD); and delay tap **3** can be cross-faded between a second odd-delayed field, i.e., a field having a relative delay of 1 field (1FD), and one of the two even-delayed fields immediately adjacent the second odd-delayed field, i.e., a field having a relative delay of 0 fields or 2 fields (0FD or 2FD). In particular, as the difference in the pixel difference values ΔPIX between an odd-delayed field and one of the immediately adjacent odd-delayed fields increases, indicating increased motion, the delay taps **1** and **3** are faded to an odd field delay. On the contrary, as the difference in the pixel difference values ΔPIX between an odd-delayed field and one of the immediately adjacent odd-delayed fields decreases, indicating decreased motion, the delay taps **1** and **3** are faded to one of the immediately adjacent even-delayed fields. To ensure that there is cross-fading between an odd-delayed field and an even-delayed field within the same scene, the lesser of the two even field delays (−2FD) immediately adjacent the first odd-delayed field (−1FD), and the lesser of the two even field delays (0FD) immediately adjacent the second odd-delayed field (1FD) are selected at the beginning of the scene, and the greater of the two even field delays (0FD) immediately adjacent the first odd-delayed field (−1FD) and the greater of the two even field delays (2FD) immediately adjacent the second odd-delayed field (1FD) are selected at the end of the scene.

[0041] Cross-fading between the first odd-delayed field (−1FD) and one of the two immediately adjacent even-

delayed fields (−2FD or 0FD) is accomplished through a first cross-fader **126** and associated first switch **130**. In particular, the 525-line delay **106** and a 262½ line delay are connected between the fixed delay **60** and a first input of the first cross-fader **126**. The 262½-line delay is created by averaging, through an averaging circuit **118**, a 262-line delay **108** with the 262-line delay **108** in series with a 1-line delay **120**. The 525-line delay **106** and the 525-line delay **106** in series with a 525-line delay **110** are connected between the fixed delay **60** and a second input of the first cross-fader **126** via the first switch **130**.

[**0042**] Similarly, cross-fading between the second odd-delayed field (1FD) and one of the two immediately adjacent even fields (0FD or 2FD) is accomplished through a second cross-fader **128** and associated second switch **132**. In particular, the **525** line delays **106** and **110** and a 262½-line delay are connected between the fixed delay **60** and a first input of the second cross-fader **128**. The 262½-line delay is created by averaging, through an averaging circuit **122**, a 262-line delay **112** with the 262-line delay **112** in series with a 1-line delay **124**. The 525-line delays **106** and **110**, and the 525 line delays **106** and **110** in series with a 525-line delay **114** are connected between the fixed delay **60** and another input of the second cross-fader **128** via the second switch **132**.

[**0043**] A switch signal SWITCH, preferably generated in the field sequence analyzer **16**, is applied to the first switch **130** to toggle selection between one of the two even field delays (−2FD or 0FD) immediately adjacent the first odd-delayed field (−1FD). The same switch signal SWITCH is applied to the second switch **132** to toggle selection between one of the two even field delays (0FD or 2FD) immediately adjacent the second odd-delayed field (1FD). Thus, application of the switch signal SWITCH to the respective first and second switches **130** and **132** either outputs the lesser of the even field delays (−2FD and 0FD) to the respective first and second cross-faders **126** and **128**, preferably at the beginning of the scene, or the greater of the even field delays (0FD and 2FD) to the respective first and second cross-faders **126** and **128**, preferably at the end of the scene. As will be described in further detail below, application of the first control signal CTRL™ to the first cross-fader **126** cross-fades between the first odd-delayed field (−1FD) and the selected immediately adjacent even field (−2FD or 0FD). Similarly, application of the second control signal CTRL**2** to the second cross-fader **128** cross-fades between the second odd-delayed field (1FD) and the selected immediately adjacent even field (0FD or 2FD).

[**0044**] The first control signal CTRL**1** is generated thusly. A first input of a first absolute value subtractor **138** is coupled to the 262-line delay **108** and a second input of the first absolute value subtractor **138** is coupled to the output of a third switch **134**. A first input of the third switch **134** is coupled to the 262-line delay **104** and a second input of the third switch **134** is coupled to the 262-line delay **112**. Thus, the first absolute value subtractor **138** compares, on a pixel-by-pixel basis, the first odd-delayed field (−1FD) with one of the fields having an odd field delay (−3FD or 1FD) immediately adjacent the first odd-delayed field (−1FD). The switch signal SWITCH is applied to the third switch **134** to select between the two immediately adjacent odd-delayed fields (−3FD or 1FD). That is, at the beginning of the scene, the lesser immediately adjacent odd-delayed field (3FD) immediately adjacent the first odd-delayed field (−1FD) is selected. At the end of the scene, the greater immediately adjacent odd-delayed field (1FD) is selected. The first absolute value subtractor **138** calculates the absolute difference between the pixels of the first odd-delayed field (−1FD) and the selected immediately adjacent odd-delayed field (3FD or 1FD), and generates an absolute pixel difference ΔPIX value in response thereto Because the determination of the motion within the immediately adjacent even-delayed field (−2FD or 0FD) is pertinent to whether the first cross-fader **126** fades to the selected immediately adjacent even-delayed field (−2FD or 0FD), the absolute pixel difference value ΔPIX is preferably further processed to provide a more accurate estimate of the motion within the selected immediately adjacent even-delayed field (−2FD or 0FD). Thus, a first input of a first fade control **140** is directly coupled to the output of the first absolute value subtractor **138**, and a second input of the first fade control **140** is coupled to the output of the absolute value subtractor **138** via a 1-line delay **142**.

[**0045**] The first fade control **140** averages, or alternatively, adds the absolute pixel difference value ΔPIX with the absolute pixel difference value ΔPIX delayed by 1-line. Alternatively, the fade control **140** selects the maximum of the absolute pixel difference value ΔPIX and the absolute pixel difference value ΔPIX delayed by 1-line, so that the first cross-fader **126** conservatively fades to the selected immediately adjacent even field delay (−2FD or 0FD). The fade control **140** generates the first control signal CTRL**1**, which is input into the first cross-fader **126**. As the amplitude of the first control signal CTRL**1** decreases, the further the first cross-fader **126** fades to the selected immediately adjacent even-delayed field (−2FD or 0FD), and as the amplitude of the first control signal CTRL**1** increases, the further the first cross-fader **126** fades to the first odd-delayed field (−1FD).

[**0046**] The second control signal CTRL**2** is generated thusly. A first input of a second absolute value subtractor **144** is coupled to the 262-line delay **112** and a second input of the second absolute value subtractor **144** is coupled to the output of a fourth switch **136**. A first input of the fourth switch **136** is coupled to the 262-line delay **108** and a second input of the fourth switch **136** is coupled to a 262-line delay **116**. Thus, the second absolute value subtractor **144** compares, on a pixel-by-pixel basis, the second odd-delayed field (1FD) with one of the odd-delayed fields (−1FD or 3FD) immediately adjacent the second odd-delayed field (1FD). The switch signal SWITCH is applied to the fourth switch **136** to select between the two immediately adjacent odd-delayed fields (−1FD or 3FD). That is, at the beginning of the scene, the lesser delayed immediately adjacent odd field (−1FD) is selected, and at the end of the scene, the greater immediately adjacent odd-delayed field (3FD) is selected. The second absolute value subtractor **144** calculates the absolute difference between the pixels of the second odd-delayed field (1FD) and the selected immediately adjacent odd field (−3FD or 1FD), and generates an absolute pixel difference ΔPIX value in response thereto. Because the determination of the motion within the immediately adjacent even-delayed field (0FD or 2FD) is pertinent to whether the second cross-fader **128** fades to the selected immediately adjacent even-delayed field (0FD or 2FD), the absolute pixel difference value ΔPIX is preferably further processed to provide a more accurate estimate of the motion within the selected immediately adjacent even-delayed field (0FD or 2FD).

Thus, a first input of a second fade control **146** is directly coupled to the output of the second absolute value subtractor **144**, and a second input of the second fade control **146** is coupled to the output of the second absolute value subtractor **144** via a 1-line delay **148**.

[**0047**] The second fade control **146** averages, or alternatively, adds the absolute pixel difference value ΔPIX with the absolute pixel difference value ΔPIX delayed by 1-line. Alternatively, the second fade control **146** selects the maximum of the absolute pixel difference value ΔPIX and the absolute pixel difference value ΔPIX delayed by 1-line, so that the first cross-fader **126** conservatively fades to the selected immediately adjacent even-delayed field (−0FD or 2FD). The second fade control **146** generates the second control signal CTRL2, which is input into the second cross-fader **128**. As the amplitude of the second control signal CTRL2 decreases, the further the second cross-fader **128** fades to the selected immediately adjacent even-delayed field (0FD or 2FD), and as the amplitude of the second control signal CTRL2 increases, the further the second cross-fader **128** fades to the second odd-delayed field (1FD).

[**0048**] The 2-3 field pattern fixer **10** can be selectively operated in the one-pass mode or the two-pass mode. Operation of the 2-3 field pattern fixer **10** in the two-pass mode is described as follows. During the first pass, the disrupted video signal $VIDEO_{IN}$ is serially received at the input **22** of the field sequence detector **12** where the fields of the discontinuous 2-3 field sequence are sequentially detected and compared to the fields two fields previous. The field sequence detector **12** generates and outputs field difference values ΔFD at the output **32** in response to these comparisons. The time code/field difference encoder **40** encodes the current time codes TC_{CUR} and the corresponding field difference values ΔFD into a series of character strings CDE_{ANL}, which are stored in the FIFO **42** for subsequent analysis by the field sequence analyzer **16**. During the first pass, the trigger signals S_{TR} are either not encoded into the character strings CDE_{ANL} or ignored by the field sequence analyzer **16**.

[**0049**] The field sequence analyzer **16** receives the series of character strings CDE_{ANL} at the input port **44**, and then decodes and analyzes the series of field difference values ΔFD to determine a discontinuity of the discontinuous 2-3 field sequence, and in particular, the first field of the scene change and phase change of the discontinuous 2-3-field sequence. Because the undisrupted video signal $VIDEO_{OUT}$ is not generated until the second pass in the two-pass mode, the field sequence analyzer **16** can analyze the entire scene to facilitate the determination of the first field of the scene change and the phase change of the discontinuous 2-3 field sequence. This capability becomes significant when the beginning of the scene has no or very little motion making it sometimes difficult to distinguish between a third field of a distinct three-field sequence 3FS and the other fields until later in the scene. Thus, although the beginning of the scene may not be determined until much later in the scene, the second pass allows the correction signal S_{COR} to be timely applied to the duplicated disrupted video signal $VIDEO_{IN}$ at the beginning of the scene. The field sequence generator **18** then generates reorganization information in response to this information. In particular, the field sequence generator **18** generates the field sequence correction signal S_{COR} and the corresponding trigger time code TC_{TR}. This information is logged into the log **48**. Several scenes can be analyzed, and thus, several correction signals S_{COR} and corresponding trigger time codes T_{CTR} can be logged into the log prior to the second pass.

[**0050**] The reorganization of the fields of the discontinuous 2-3 field sequence into the fields of the continuous field sequence typically results in continuous 2-3 field sequence that is distorted in time, causing the resulting video signal and the corresponding audio signal, which is not distorted in time, to be out of synchronization. This will typically not create a problem when the respective video and audio signals are unsynchronized by a minimal amount of fields, which may result from a delay through a single scene change. For example, edit point EP depicted in **FIG. 2** causes a 1 field delay for the film frames represented in the first continuous 2-3 field sequence (e.g., the partial sequence **O2/O1** in the first continuous 2-3 field sequence is one field delayed from the partial sequence **O1/O2** in the discontinuous 2-3 field sequence) and a −1½ field delay for the film frames represented in the second continuous 2-3 field sequence (e.g., the partial sequence **O1/O2/O1** in the second continuous 2-3 field sequence is −1½ field delayed from the partial sequence **O1/O2** in the discontinuous 2-3 field sequence).

[**0051**] These delays are relatively minimal. If left unchecked, however, the cumulative field delay through a series of scene changes could be relatively great, resulting in the de-synchronization of the respective video and audio signals. As such, for each scene change, the field sequence generator **18** selects a particular delay sequence that would minimize the cumulative field delay, or at least maintain the cumulative field delay within a certain range. Typically, this delay range will include values of −2, −1½, −1, −½, 0, +½, +1, +1½, +2. The delay range, however, can be greater or less, depending on the tolerance of the delay between the respective video and audio signals

[**0052**] Also, in selecting particular delay sequences, the field sequence generator **18** preferably discards disjoined single field frames, such that the resulting delay sequence does not include an odd field delay. For example, the **M1** field depicted in **FIG. 2** is discarded in generating the second continuous 2-3 field sequence. Thus, in selecting particular field delay sequences, the field sequence generator **18** attempts to maintain the cumulative field delay within a defined range, while discarding as many disjoined single field frames as possible. In the two-pass mode, several scenes can be analyzed at a time, thereby allowing a more efficient selection of the delay sequences. In the one-pass mode, the field sequence generator **18** must determine the delay sequence as each scene is analyzed, and therefore must determine whether a disjoined single field frame should be discarded based on the cumulative field delay at that point. Preferably, while in the one-pass mode, the cumulative field delay is maintained on the positive side of the range to allow the field sequence generator **18** more flexibility in discarding single field frames. Of course, under certain circumstances, whether in the two-pass mode or the one-pass mode, the cumulative field delay becomes too negative if every disjoined single field frame is discarded.

[**0053**] For example, if the cumulative field delay is negative, such as, e.g., −2 fields, and the tolerance level of the

delay between the respective video and audio signals is low, the field sequence generator **18** may select the delay sequence used to generate the first continuous 2-3 field sequence in **FIG. 2**, so that the absolute cumulative field delay is lessened by 1 field. Although this results in the blurring of the M2 field, selection of the delay sequence used to generate the second continuous 2-3 field sequence would have increased the absolute cumulative field delay by 1½ fields to a cumulative total of −3½ fields, which may produce a result more undesirable than that produced by selection of the delay sequence used to generate the first continuous 2-3 field sequence.

[**0054**] Thus, under certain circumstances, the field sequence generator **18** must select between which of the disjoined single field frames to discard. This selection process is preferably based on detected motion between the disjoined single field frame and the immediately adjacent frame (subsequent frame if at the beginning of the scene, and preceding frame if at the end of the scene), which can be obtained from the field difference values ΔFD analyzed by the field sequence analyzer **16**. In the case where the basic bank of delay taps **62** is employed, the disjoined single field frames in which there is detected relatively little motion are preferably discarded before those in which there is detected relatively great motion. This is because the vertical blurring will not be as noticeable to the viewer when there is a lot of motion in the vertically blurred frame. In the case where the compensating bank of delay taps **102** is employed, the disjoined single field frames in which there is detected relatively great motion are preferably discarded before those in which there is detected relatively great motion. This is because the compensating bank of delay taps **102** cannot compensate for vertically blurred frames that have relatively a large amount of motion.

[**0055**] During the second pass, a duplicated disrupted video signal $VIDEO_{IN}'$ is received at the first input **55** of the multiple delay tap circuit **20**. At the same time or prior thereto, the field sequence generator **18** encodes the multitude of correction signals S_{COR} and corresponding trigger time codes TC_{TR} received from the log **48** into a corresponding multitude of control character strings CDE_{CTRL}. These control character strings CDE_{CTRL} are then output at the output port **46** into the FIFO **50**. The first control character string CDE_{CTRL} is advanced into and decoded by the time code/correction signal decoder **52**, which then transmits the field sequence correction signal S_{COR} to the delay tap selector **64** and the trigger time code TC_{TR} to the second input **36** of the time code comparator **14**. The current time code TC_{CUR}' of the duplicated disrupted video signal $VIDEO_{IN}'$ is input into the first input **34** of the time code comparator **14**. The current time code TC_{CUR}' and the trigger signals S_{TR} are encoded by the time code/field difference encoder **40** into a series of character strings CDE_{ANL} for use by the PC to provide the user status. During the second pass, field difference values ΔFD are either not encoded into the character strings CDE_{ANL} or ignored by the field sequence analyzer **16**.

[**0056**] The field sequence correction signal S_{COR} is not applied to the duplicated disrupted video signal $VIDEO_{IN}'$ until the current time code TC_{CUR}' of the duplicated disrupted video signal $VIDEO_{IN}'$ matches the trigger time code TC_{TR}. In this connection, the time code comparator **14** compares the current time code TC_{CUR}' to the trigger time code TC_{TR} and generates the trigger signal S_{TR} in response thereto. The trigger signal S_{TR} is input into the FIFO **50** and the register **82** and counter **86** of the multiple delay tap circuit **20**. If the current time code TC_{CUR}' is less than the trigger time code TC_{TR}, the first bit of the two bit trigger signal S_{TR} is low. In this case, the FIFO **50** is not advanced, the register does not save the field sequence correction signal S_{COR}, and the counter is not reset to "1".

[**0057**] If the current time code TC_{CUR} is equal to the trigger time code TC_{TR}, the first bit of the two bit trigger signal S_{TR} is high. In this case, the register **82** of the multiple delay tap circuit **20** saves the field sequence correction signal S_{COR}, and the counter **86** is reset to "1". The field sequence correction signal S_{COR} is output from the register **82** to the switch **84**. With the counter **86** set at "1", the first delay number is sent to the delay control circuit **88**, which selects the delay tap within the bank of delay taps **62** in accordance with the delay number. The FIFO **50** is advanced, transmitting the next control character string CDE_{CTRL} to the time code/correction signal decoder **52**, which decodes and transmits the next field sequence correction signal S_{COR} to the multiple delay tap circuit **20** and the next trigger time code TC_{TR} to the second input **36** of the time code comparator **14**.

[**0058**] Until the current time code TC_{CUR} matches the next trigger time code TC_{TR}, the current field sequence correction signal S_{COR} is applied to the duplicated disrupted video signal $VIDEO_{IN}'$, with the counter **86** counting from 1 to 5. In this manner, the delay numbers of the current field sequence correction signal S_{COR} are sequentially sent to the delay control circuit **88**, thereby sequentially delaying the selected fields of the duplicated disrupted video signal $VIDEO_{IN}'$ to generate the undisrupted video signal $VIDEO_{OUT}$ at the output **56**. Thus, the correction signal S_{COR} is applied to the duplicated disrupted video signal $VIDEO_{IN}'$ in real time, i.e., at a field rate equal to the field rate of the disrupted video signal $VIDEO_{IN}$.

[**0059**] Application of the delay numbers −2, −1, 0, 1 and 2 to the delay control circuit **88** prompts respective selection of the corresponding delay taps **0**, **1**, **2**, **3** and **4**, and thus selection of a distinct field of the duplicated disrupted video signal $VIDEO_{IN}'$ that has a delay corresponding to the current delay number. If the compensating bank of delay taps **102** depicted in **FIG. 4** are used, selection of the fields of the duplicated disrupted video signal $VIDEO_{IN}'$ that have been delayed an odd number of fields are not distinct, but are rather cross-faded with an immediately adjacent field. In particular, application of an odd delay number to the delay control circuit **88** prompts selection of the corresponding odd delay tap, and thus selection of the corresponding odd-delayed field and one of the immediately adjacent even-delayed fields. At the same time that the odd delay number is applied to the delay control circuit **88**, the switch signal SWITCH is sent from the field sequence analyzer **16** to the first, second, third and fourth switches **130**, **132**, **134** and **136** to select the particular even-delayed field to and from which the odd-delayed field is cross-faded. Thus, if at the beginning of the scene, the switch signal SWITCH causes the switches **130**, **132**, **134** and **136** to switch down to select the greater of the immediately adjacent even-delayed fields (0FD and 2FD). If at the end of the scene, the switch signal SWITCH causes the switches **130**, **132**, **134**

and **136** to switch up to select the lesser of the immediately adjacent even-delayed fields (2FD and 0FD).

[**0060**] For instance, if delay tap **1** is selected and the switches **130** and **134** are switched down, the first odd-delayed field (−1FD) and the greater immediately adjacent even field delay (0FD) are cross-faded. The first absolute value subtractor **138** compares the pixels of the first odd-delayed field (−1FD) with the pixels of the greater immediately adjacent odd-delayed field (1FD), generating the pixel difference value ΔPIX. The pixel difference value ΔPIX is processed through the first fade control **140**, generating the first control signal CTRL**1**, which is applied to the first cross-fader **126**. The first cross-fader **126** then cross-fades between the first odd-delayed field (−1FD) and the greater immediately adjacent even-delayed field (0FD) in accordance with the first control signal CTRL**1**. If the switches **130** and **134** are switched up, the first odd-delayed field (−1FD) and the lesser immediately adjacent even-delayed field (−2FD) are cross-faded. The first absolute value subtractor **138** compares the pixels of the first odd-delayed field (−1FD) with the pixels of the lesser immediately adjacent odd-delayed field (−3FD), generating the pixel difference value ΔPIX. The pixel difference value ΔPIX is processed through the first fade control **140**, generating the first control signal CTRL**1**, which is applied to the first cross-fader **126**. The first cross fader **126** then cross-fades between the first odd-delayed field (−1FD) and the lesser immediately adjacent even-delayed field (−2FD) in accordance with the first control signal CTRL**1**.

[**0061**] In a similar manner, if delay tap **3** is selected and the switches **132** and **136** are switched down, the second odd-delayed field (1FD) and the greater immediately adjacent even-delayed field (2FD) are cross-faded. If the switches **132** and **136** are switched up, the second odd-delayed field (1FD) and the lesser immediately adjacent even-delayed field (0FD) are cross-faded.

[**0062**] Referring back to **FIG. 1**, when the current time code TC_{CUR} does match the next trigger time code T_{CTR}, the first bit of the two bit trigger signal S_{TR} again is high, thereby saving the next field sequence correction signal S_{COR} in the register **82** and clearing the counter **86** to apply the next field sequence correction signal S_{COR} to the duplicated disrupted video signal $VIDEO_{IN}$'. The FIFO **50** is advanced to input the next control character string CDE_{CTRL} into the time code/correction signal decoder **52** for transmission of the next field sequence correction signal S_{COR} to the multiple delay tap circuit **20** and next trigger time code T_{CTR} to the time code comparator **14**. This process is repeated until the last field sequence correction signal S_{COR} has been applied to the duplicated disrupted video signal $VIDEO_{IN}$'.

[**0063**] Operation of the 2-3 field pattern fixer **10** in the one-pass mode is described as follows. The disrupted video signal $VIDEO_{IN}$ is serially received at the input **22** of the field sequence detector **12** and the first input **55** of the multiple delay tap circuit **20**. The field sequence detector **12** sequentially detects the fields of the discontinuous 2-3 field sequence and compares these fields to the fields two fields previous. The field sequence detector **12** generates and outputs field difference values ΔFD at the output **32** in response to these comparisons. The time code/field difference encoder **40** encodes the current time codes TC_{CUR}, the corresponding field difference values ΔFD, and the trigger signals S_{TR} from the time code comparator **14** into a series of character strings CDE_{ANL}, which are stored in the FIFO **42** for subsequent analysis by the field sequence analyzer **16**. When the 2-3 field pattern fixer **10** is in the one-pass mode, the trigger signals S_{TR} are either not encoded in the character strings CDE_{ANL} or are ignored by the field sequence analyzer **16**.

[**0064**] The field sequence analyzer **16** receives the series of character strings CDE_{ANL} at the input port **44**, and then decodes and analyzes the series of field difference values ΔFD to determine a discontinuity of the discontinuous 2-3 field sequence, and in particular, the first field of the scene change and phase change of the discontinuous 2-3 field sequence. The field sequence generator **18** then generates reorganization information in response to this information. In particular, the field sequence generator **18** generates the field sequence correction signal S_{COR}. A corresponding trigger time code TC_{TR} is not generated, since the field sequence correction signal S_{COR} must be applied to the disrupted video signal $VIDEO_{IN}$ as it is generated. The fixed delay **60** in the multiple delay tap circuit **20** is long enough to allow the sequence analyzer **16** and field sequence generator **18** to analyze the series of field difference values ΔFD and generate the field sequence correction signal S_{COR} in time to apply the field sequence correction signal S_{COR} to the selected fields of the disrupted video signal $VIDEO_{IN}$. If the disrupted video signal $VIDEO_{IN}$ originates from a digital recording, the identity or near identity of the first and third fields of each distinct three field sequence 3FS should allow the field sequence analyzer **16** to determine the discontinuity of the discontinuous 2-3 field sequence quickly, even if there is no or very little motion within the scene.

[**0065**] The field sequence generator **18** encodes the field sequence correction signal S_{COR} into a control character string CDE_{CTRL}, which is then output at the output port **46** into the FIFO **50**. The control character string CDE_{CTRL} is advanced into and decoded by the time code/correction signal decoder **52**, which then transmits the field sequence correction signal S_{COR} to the multiple delay tap circuit **20**, where it is applied to the disrupted video signal $VIDEO_{IN}$. In this connection, the register **82** of the multiple delay tap circuit **20** saves the field sequence correction signal S_{COR}, and the counter **86** is reset to "1" upon the receipt of a trigger signal, which may be generated in the PC or timing circuitry. The field sequence correction signal S_{COR} is output from the register **82** to the switch **84**. With the counter **86** set at "1", the first delay number is sent to the delay control circuit **88**, which selects the delay tap within the bank of delay taps **62** in accordance with the delay number. Until the field sequence generator **18** generates another field sequence correction signal S_{COR}, the current field sequence correction signal S_{COR} is applied to the duplicated disrupted video signal $VIDEO_{IN}$', with the counter **86** counting from 1 to 5. In this manner, the delay numbers of the current field sequence correction signal S_{COR} are sequentially sent to the delay control circuit **88**, thereby sequentially delaying the selected fields of the disrupted video signal $VIDEO_{IN}$ to generate the undisrupted video signal $VIDEO_{OUT}$ at the output **56**. Thus, the correction signal S_{COR} is applied to the disrupted video signal $VIDEO_{IN}$ in real time, i.e., at a field rate equal to the field rate of the disrupted video signal $VIDEO_{IN}$.

[0066] It should be noted that although the 2-3 field pattern fixer **10** is shown as being selectively operated in a one-pass mode and a two-pass mode, it will be readily apparent to those skilled in the art that a 2-3 field pattern fixer that is able to operate in only one of the modes can be constructed without straying from the principles taught by this invention. For example, a 2-3 field sequence fixer that only operates in a one-pass mode obviates the need for circuitry that is otherwise required for the 2-3 field sequence fixer to operate in the two-pass mode. Furthermore, operation of the 2-3 field sequence fixer solely in the one-pass lends itself to a strict hardware implementation of the 2-3 field sequence fixer, eliminating the need for a PC. In this connection, the field sequencer **16** and field sequence generator **18** can be implemented in hardware, and the timecode comparator **14**, timecode/field difference encoder **40**, FIFO's **42** and **50**, timecode/correction signal decoder **52** and reset decoder **54** can be eliminated. In this case, the field difference value ΔFD can be input directly into the field sequence analyzer **16** and the field sequence correction signal S_{COR} can be output directly from the field sequence generator **18**.

[0067] While preferred methods and embodiments have been shown and described, it will be apparent to one of ordinary skill in the art that numerous alterations may be made without departing from the spirit or scope of the invention. Therefore, the invention is not to be limited except in accordance with the following claims.

What is claimed:

1. A method of generating an undisrupted video signal comprising a series of fields arranged in a continuous 2-3 field sequence, the method comprising:

receiving a disrupted video signal, the disrupted video signal comprising a series of fields arranged in a discontinuous 2-3 field sequence;

analyzing the discontinuous 2-3 field sequence of the disrupted video signal to determine a discontinuity;

generating a correction signal based on the determined discontinuity; and

generating the undisrupted video signal by applying the correction signal to a video signal.

2. The method of claim 1, wherein the undisrupted video signal is generated by rearranging the series of fields in one of the disrupted video signal and a duplicate of the disrupted video signal in accordance with the correction signal.

3. The method of claim 1, wherein the correction signal is applied to the disrupted video signal.

4. The method of claim 3, wherein the received disrupted video signal is delayed at least an amount of time required to generate the correction signal.

5. The method of claim 3, wherein the correction signal is applied to the video signal as the correction signal is generated.

6. The method of claim 1, wherein the correction signal is applied to a duplicate of the disrupted video signal.

7. The method of claim 6, further comprising:

determining a plurality of discontinuities in the discontinuous 2-3 field sequence;

determining a first plurality of time codes within the disrupted video signal, the first plurality of time codes respectively associated with the plurality of discontinuities;

generating a plurality of correction signals respectively based on the plurality of discontinuities;

receiving the duplicated disrupted video signal;

determining a second plurality of time codes within the duplicated disrupted video signal, the second plurality of time codes corresponding with the first plurality of time codes; and

generating the undisrupted video signal by sequentially applying the plurality of correction signals to the duplicated disrupted video signal as the corresponding second plurality of time codes are determined.

8. The method of claim 1, wherein the correction signal is generated by selecting a series of delays, each delay applied to one field of either the disrupted video signal or a duplicate of the disrupted video signal.

9. The method of claim 8, wherein a cumulative field delay is generated within the undisrupted video signal, and wherein selection of the series of delays is based on a minimization of the cumulative field delay.

10. The method of claim 9, wherein the disrupted video signal includes a plurality of disjoined single field frames, and wherein selection of the series of delays is based on discarding of the disjoined single field frames.

11. The method of claim 10, wherein the discarding of disjoined single field frames are based on a relative motion detected within the disjoined single field frames.

12. The method of claim 11, wherein the disjoined single field frames having a relatively great motion are discarded.

13. The method of claim 11, wherein the disjoined single field frames having relatively little motion are discarded.

14. The method of claim 1, wherein an edit point and a phase change within the discontinuous 2-3 field sequence are determined, and the generation of the correction signal is based on the detected phase change and edit point.

15. The method of claim 14, wherein the edit point and a phase of the discontinuous 2-3 field sequence are determined by sequentially comparing fields twice removed in sequence.

16. The method of claim 1, wherein a plurality of correction signals are generated and sequentially applied to the video signal.

17. The method of claim 1, wherein the application of the plurality of correction signals to the video signal generates a cumulative field delay between the discontinuous 2-3 field sequence of the disrupted video signal and the continuous 2-3 field sequence of the undisrupted video signal, and wherein the plurality of correction signals are generated, such that the cumulative field delay is maintained within a range of selected values.

18. A method of correcting a disrupted video signal comprising a series of fields arranged in a discontinuous 2-3 field sequence, the method comprising:

receiving the disrupted video signal;

analyzing the discontinuous 2-3 field sequence of the disrupted video signal to determine a discontinuity;

generating a correction signal based on the determined discontinuity; and

generating the undisrupted video signal by applying the correction signal to a video signal at a field rate equal to the field rate of the disrupted video signal.

19. The method of claim 18, wherein the undisrupted video signal is generated by rearranging the series of fields in either the disrupted video signal or the duplicate of the disrupted video signal in accordance with the correction signal.

20. The method of claim 18, wherein the correction signal is applied to the disrupted video signal.

21. The method of claim 18, wherein the correction signal is applied to a duplicate of the disrupted video signal.

22. The method of claim 21, further comprising:

determining a plurality of discontinuities in the discontinuous 2-3 field sequence;

detecting a first plurality of time codes within the disrupted video signal, the first plurality of time codes respectively associated with the plurality of discontinuities;

generating a plurality of correction signals respectively based on the plurality of discontinuities;

receiving the duplicated disrupted video signal;

detecting a second plurality of time codes within the duplicated disrupted video signal, the second plurality of time codes corresponding with the first plurality of time codes; and

generating the undisrupted video signal by sequentially applying the plurality of correction signals to the duplicated disrupted video signal as the corresponding second plurality of time codes are detected, each of the correction signals being applied to the duplicated disrupted video signal at a field rate equal to a field rate of the disrupted video signal.

23. The method of claim 18, wherein an edit point and a phase change within the discontinuous 2-3 field sequence are determined, and the generation of the correction signal is based on the detected phase change and edit point.

24. The method of claim 18, wherein a plurality of correction signals are generated and sequentially applied to the video signal, each of the correction signals being applied to the duplicated disrupted video signal at a field rate equal to a field rate of the disrupted video signal.

25. A 2-3 field pattern fixer, comprising:

a field sequence detector configured for receiving a disrupted video signal having a discontinuous 2-3 field sequence and generating a series of field difference values;

a field sequence analyzer having an input coupled to an output of the field sequence detector and being configured for generating field sequence information based on the series of field difference values;

a field sequence generator having an input coupled to an output of the field sequence analyzer and being configured for generating a sequence of delay numbers based on the field sequence information; and

a multiple delay tap circuit having an input coupled to an output of the field sequence generator and being configured for receiving a video signal and for applying the sequence of delay numbers thereto to generate an undisrupted video signal having a continuous 2-3 field sequence.

26. The 2-3 field pattern fixer of claim 25, wherein the field sequence analyzer and the field sequence generator are embodied in a computer.

27. The 2-3 field pattern fixer of claim 25, wherein one of the field sequence generator and field sequence analyzer is further configured to receive a current time code of the disrupted video signal and generate a trigger time code in response thereto;

wherein the 2-3 field pattern fixer further comprises a time code comparator having an first input coupled to an output of the one of the field sequence generator and field sequence analyzer and a second input for receiving a current time code of the duplicated disrupted video signal, the time code comparator configured for generating a trigger signal; and

wherein the multiple delay tap circuit includes a second input coupled to an output of the time code comparator, and configured for initially applying the sequence of delay numbers to a duplicated of the disrupted video signal when the trigger signal equals a selected value.

28. The 2-3 field pattern fixer of claim 27, further comprising a First-In First-Out (FIFO) memory device coupled between the field sequence generator and the multiple delay tap circuit; and

wherein the field sequence generator is configured for generating and storing a series of correction signals within the FIFO.

29. The 2-3 field pattern fixer of claim 25, wherein the 2-3 field pattern fixer can be selectively operated in a one-pass mode and a two-pass mode.

30. The 2-3 field pattern fixer of claim 25, wherein the multiple delay tap circuit is implemented using RAM addressing circuitry.

31. The 2-3 field pattern fixer of claim 25, wherein the multiple delay tap circuit comprises a bank of delay taps, and a delay tap selector coupled to the bank of delay taps for selecting one of the delay taps.

32. The 2-3 field pattern fixer of claim 25, wherein the bank of delay taps comprises an odd delay tap with a cross-fader.

33. A method of generating an undisrupted video signal comprising a series of fields arranged in a continuous 2-3 field sequence, the method comprising:

receiving a disrupted video signal, the disrupted video signal comprising a series of fields arranged in a discontinuous 2-3 field sequence;

analyzing the discontinuous 2-3 field sequence of the disrupted video signal to determine a discontinuity within the discontinuous **23** field sequence;

selecting a series of field delays based on the determined discontinuity;

applying each field delay to one field of either the disrupted video signal or a duplicate of the disrupted video signal to generate the undisrupted video signal, wherein a field is delayed an odd number of fields; and

cross fading between the odd-delayed field and another field to generate a field within the disrupted video signal.

34. The method of claim 33, wherein the cross-fading is between the odd-delayed field and an even-delayed field.

35. The method of claim 34, wherein the odd-delayed field is at the beginning of a scene, and wherein the cross-fading is between the odd-delayed field and an even-delayed field immediately subsequent to the odd-delayed field.

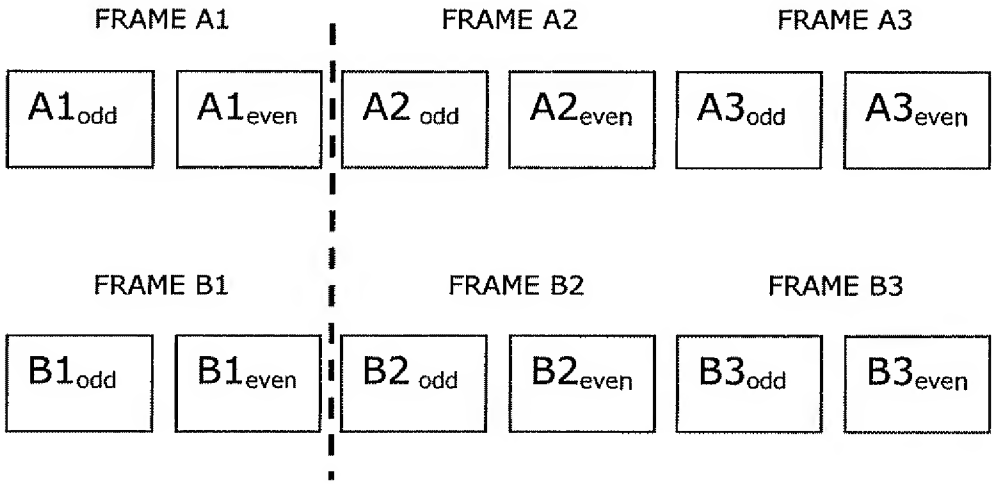
36. The method of claim 34, wherein the odd-delayed field is at the end of a scene, and wherein the cross-fading is between the odd-delayed field and an even-delayed field immediately preceding the odd-delayed field.

37. The method of claim 34, further comprising determining an absolute pixel difference between the odd delayed field and an immediately adjacent odd-delayed field, and wherein the odd-delayed field is faded to if the absolute pixel difference is relative large and the odd-delayed field is faded from if the absolute pixel difference is relatively small.

38. The method of claim 37, further comprising delaying the absolute pixel difference by one line and processing the absolute pixel difference and the delayed absolute pixel difference to more accurately estimate an absolute pixel difference between the odd-delayed field and an immediately adjacent even-delayed field.

* * * * *

DIAGRAM 1A - Correctly positioned cut from Scene A to Scene B



Output with field dominance preserved

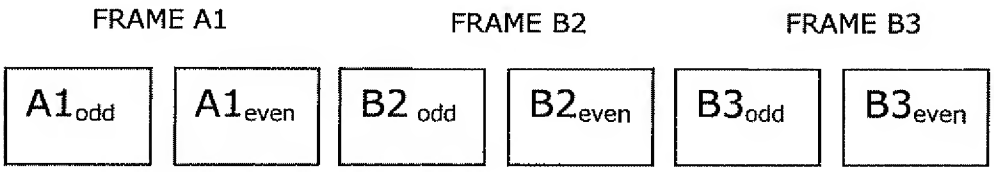
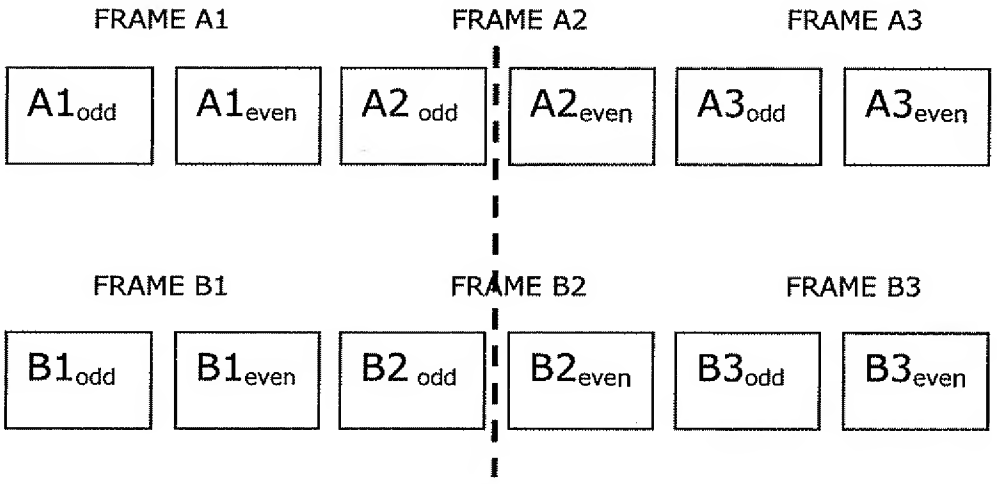


Diagram 1B - Incorrectly positioned cut from Scene A to Scene B



Output with field dominance corrupted

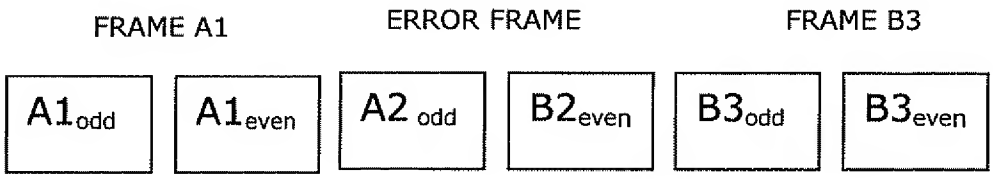
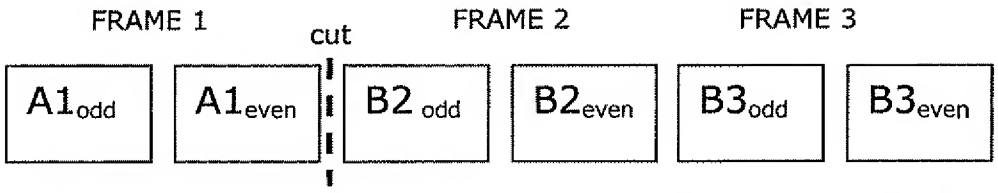
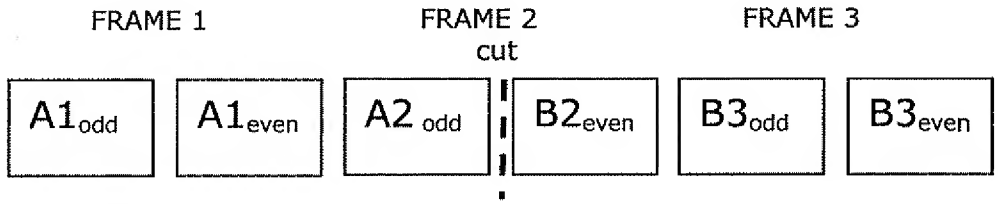


DIAGRAM 2 A



Identified cut <u>does</u> occur at frame boundary (between FRAME 1 and FRAME 2)

DIAGRAM 2 B



Identified cut <u>does not</u> occur at frame boundary

DIAGRAM 3 A

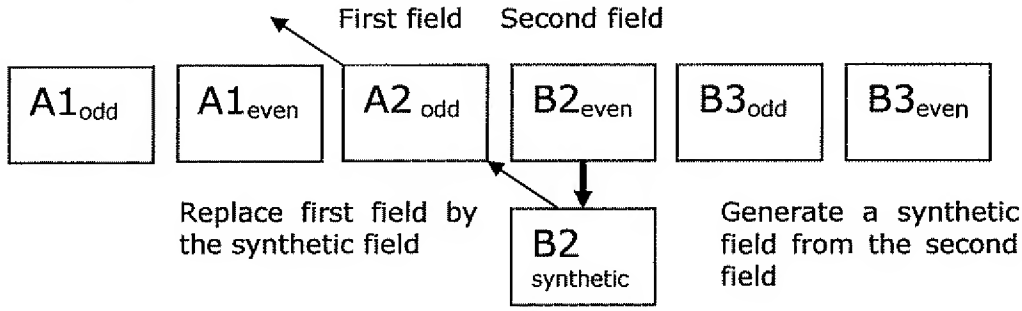
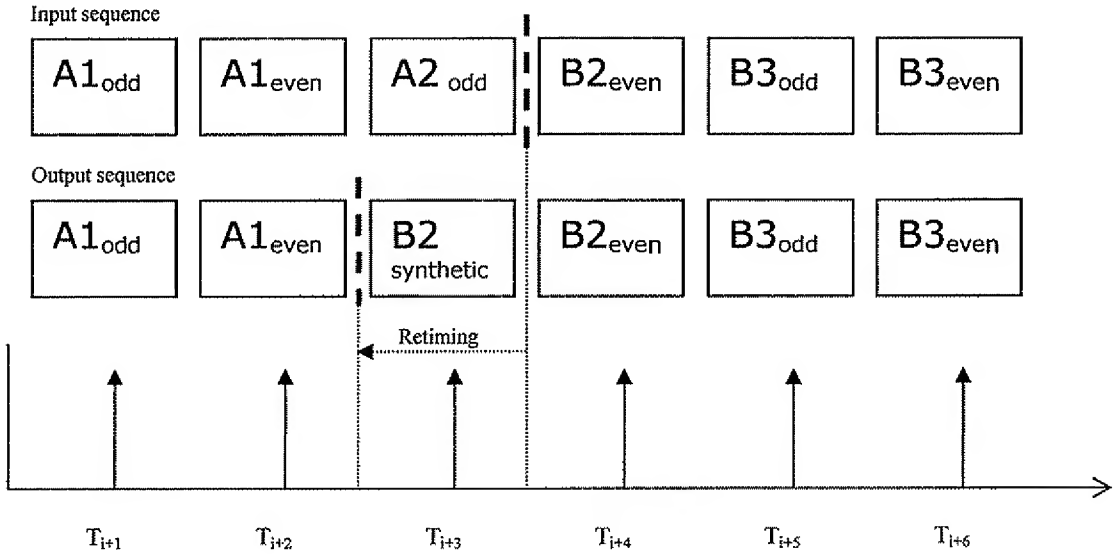


DIAGRAM 3 B

DIAGRAM 4


Input sequence
A1odd
A1even
A2 odd
B2even
B3odd
B3even
Output sequence
A1odd
A1even
B2
synthetic
B2even
B3odd
B3even
Retiming
Ti+1
Ti+2
Ti+3
Ti+4
Ti+5
Ti+6
TIME